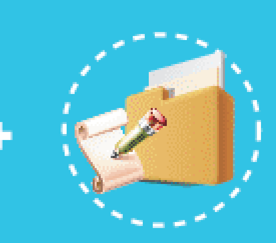
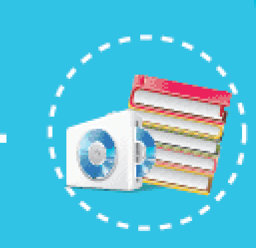
网站开发案例课堂



HTML 5 网页设计案例课堂（第2版）

刘春茂　编著

本书大量赠送资源

1 本书实例**源代码**
2 精美教学**幻灯片**
3 精品教学**视频讲座**
4 12部网页开发必备**参考手册**
5 88个实用类**网页模板**
6 100个精选的JavaScript超炫案例
7 各行业CSS+DIV布局赏析案例
8 各行业精彩**网站配色**方案赏析
9 各行业**网页样式**与**布局**案例赏析
10 Web工程师常见**面试题**

清华大学出版社

网站开发案例课堂

HTML 5 网页设计案例课堂
(第 2 版)

刘春茂　编　著

清华大学出版社
北　京

内 容 简 介

本书以零基础讲解为宗旨，用实例引导读者深入学习，采取【基础入门→核心技术→高级技能→移动开发→综合案例实战】的讲解模式，深入浅出地讲解 HTML 5 的各项技术及实战技能。

本书第Ⅰ篇【基础入门】主要内容包括新一代 Web 前端技术 HTML 5、HTML 5 网页的文档结构、HTML 5 与 HTML4 的区别等；第Ⅱ篇【核心技术】主要内容包括设计网页文本内容、设计网页列表与段落、HTM L5 网页中的图像、使用 HTML 5 建立超链接、使用 HTML 5 创建表单、使用 HTML 5 创建表格、HTML 5 中的音频和视频、使用 HTML 5 绘制图形等；第Ⅲ篇【高级技能】主要内容包括 HTML 5 中的文件与拖放、定位地理位置技术、Web 存储和通信技术、处理线程和服务器发送事件、构建离线的 Web 应用等；第Ⅳ篇【移动开发】主要内容包括 jQuery Mobile 基础、jQuery Mobile UI 组件、jQuery Mobile 事件、数据存储和读取技术等；第Ⅴ篇【综合案例实战】主要内容包括制作休闲娱乐类网页、制作企业门户类网页、制作电子商务类网页、开发连锁酒店订购系统。本书赠送了 13 个超值的王牌资源。

本书适合任何想学习网页前台设计与布局的人员，无论您是否从事计算机相关行业，无论您是否接触过 HTML 5，通过学习均可快速掌握网页的设计方法和技巧。

图书在版编目(CIP)数据

HTML5 网页设计案例课堂/刘春茂编著. —2 版. —北京：清华大学出版社，2018
(网站开发案例课堂)
ISBN 978-7-302-48916-0

Ⅰ. ①H…　Ⅱ. ①刘…　Ⅲ. ①超文本标记语言—程序设计　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 287981 号

责任编辑：张彦青
装帧设计：李　坤
责任校对：吴春华
责任印制：李红英
出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　张：**28　　**字　数：**680 千字
版　　次：2016 年 1 月第 1 版　2018 年 1 月第 2 版　　**印　次：**2018 年 1 月第 1 次印刷
印　　数：1～3000
定　　价：69.00 元

产品编号：077891-01

前　言

“网站开发案例课堂”系列图书是专门为办公技能和网页设计初学者量身定制的一套学习用书。整套书涵盖高效办公、网站开发、数据库设计等方面。整套书具有以下特点。

前沿科技

无论是网站建设、数据库设计还是 HTML 5、CSS 3，我们都精选较为前沿或者用户群最大的领域推进，帮助大家认识和了解最新动态。

权威的作者团队

组织国家重点实验室和资深应用专家联手编著该套图书，融合丰富的教学经验与优秀的管理理念。

学习型案例设计

以技术的实际应用过程为主线，全程采用图解和同步多媒体结合的教学方式，生动、直观、全面地剖析使用过程中的各种应用技能，降低难度，提升学习效率

为什么要写这样一本书

随着用户对页面体验要求的提高，页面前端技术日趋重要。HTML 5 技术成熟，在前端技术中突显优势，在各大浏览器厂商的支持下，将会更加盛行，因此本书致力于帮助读者完全掌握该技术。本书可以让读者掌握目前流行的最新前端技术，使前端从外观上变得更炫、技术上更简易。通过本书的案例实训，大学生可以很快地掌握流行的工具，提高职业化能力，从而帮助解决公司与求职者的双重需求问题。

本书特色

- 零基础、入门级的讲解

无论您是否从事计算机相关行业，无论您是否接触过网页制作和设计，都能从本书中找到最佳起点。

- 超多、实用、专业的范例和项目

本书在编排上紧密结合深入学习网页制作技术的先后过程，从 HTML 5 的基本概念开始，引导读者逐步深入地学习各种应用技巧，并侧重实战技能，使用简单易懂的实际案例进行分析和操作指导，让读者读起来简明轻松，操作起来有章可循。

- 随时检测自己的学习成果

每章首页中，均提供了学习目标，以指导读者重点学习及学后检查。

大部分章节最后的“跟我学上机”板块，均根据本章内容精选而成，读者可以随时检测

自己的学习成果和实战能力，做到融会贯通。

- 细致入微、贴心提示

本书在讲解过程中，在各章使用了“注意”“提示”“技巧”等小贴士，使读者在学习过程中更清楚地了解相关操作、相关概念，并轻松掌握各种操作技巧。

- 专业创作团队和技术支持

本书由千谷高新教育中心编著和提供技术支持。

您在学习过程中遇到任何问题，可加入 QQ 群(案例课堂 VIP)451102631 进行提问，专家人员会在线答疑。

超值资源大放送

- 全程同步教学录像

涵盖本书所有知识点，详细讲解每个实例及项目的过程及技术关键点。比看书更轻松地掌握书中所有的网页制作和设计知识，而且扩展的讲解部分可使您得到比书中更多的收获。

- 超多容量王牌资源



赠送大量王牌资源，包括实例源代码、教学幻灯片、本书精品教学视频、88 个实用类网页模板、12 部网页开发必备参考手册、HTML 5 标签速查手册、精选的 JavaScript 实例、CSS 3 属性速查表、JavaScript 函数速查手册、CSS+DIV 布局赏析案例、精彩网站配色方案赏析、网页样式与布局案例赏析、Web 前端工程师常见面试题等。读者可以通过 QQ 群(案例课堂 VIP)451102631 获取赠送资源，也可以扫描二维码，下载本书资源。

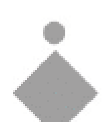
读者对象

- 没有任何网页设计基础的初学者。
- 有一定的 HTML 5 基础，想精通网页制作和设计的人员。
- 有一定的 HTML 5 基础，没有项目经验的人员。
- 正在进行毕业设计的学生。
- 大专院校及培训学校的老师和学生。

创作团队

本书由刘春茂编著，参加编写的人员还有刘玉萍、张金伟、蒲娟、周佳、付红、李园、郭广新、侯永岗、王攀登、刘海松、孙若淞、王月娇、包慧利、陈伟光、胡同夫、王伟、展娜娜、李琪、梁云梁和周浩浩。在编写过程中，我们力尽所能地将最好的讲解呈现给读者，但也难免有疏漏和不妥之处，敬请不吝指正。若您在学习中遇到困难或疑问，或有何建议，可写信至信箱 357975357@qq.com。

编　者



目　　录

第 I 篇　基 础 入 门

第 1 章　新一代 Web 前端技术 HTML 5 3

1.1　HTML 的基本概念 4

1.1.1　HTML 的发展历程 4

1.1.2　什么是 HTML 4

1.1.3　HTML 5 文件的基本结构 5

1.2　HTML 5 的优势 5

1.2.1　解决了跨浏览器问题 5

1.2.2　新增了多个新特性 5

1.2.3　用户优先的原则 6

1.2.4　化繁为简的优势 7

1.3　HTML 5 网页的开发环境 7

1.3.1　案例 1——使用记事本手工编写 HTML 5 7

1.3.2　案例 2——使用 Dreamweaver CC 编写 HTML 文件 8

1.4　使用浏览器查看 HTML 5 文件 11

1.4.1　案例 3——查看页面效果 11

1.4.2　案例 4——查看源文件 12

1.5　高手解惑 12

第 2 章　HTML 5 网页的文档结构 13

2.1　HTML 5 文件的基本结构 14

2.1.1　HTML 5 页面的整体结构 14

2.1.2　HTML 5 新增的结构标记 14

2.2　HTML 5 基本标记详解 15

2.2.1　文档类型说明 15

2.2.2　HTML 标记 15

2.2.3　头标记 head 16

2.2.4　网页的主体标记 body 18

2.2.5　页面注释标记<!-- --> 19

2.3　HTML 5 语法的变化 20

2.3.1　标签不再区分大小写 20

2.3.2　允许属性值不使用引号 20

2.3.3　允许部分属性值的属性省略 20

2.4　必知必会——HTML 5 代码规范 21

2.5　综合案例——符合 W3C 标准的 HTML 5 网页 22

2.6　跟我学上机——简单的 HTML 5 网页 22

2.7　高手解惑 23

第 3 章　HTML 5 与 HTML 4 的区别 25

3.1　新增的主体结构元素 26

3.1.1　案例 1——section 元素的使用 26

3.1.2　案例 2——article 元素的使用 26

3.1.3　案例 3——aside 元素的使用 29

3.1.4　案例 4——nav 元素的使用 31

3.1.5　案例 5——time 元素的使用 32

3.2　新增的非主体结构元素 34

3.2.1　案例 6——header 元素的使用 34

3.2.2　案例 7——hgroup 元素的使用 34

3.2.3　案例 8——footer 元素的使用 36

3.2.4　案例 9——figure 元素的使用 37

3.2.5　案例 10——address 元素的使用 39

3.3　新增其他常用元素 40

3.3.1　案例 11——mark 元素的使用 40

3.3.2　案例 12——rp 元素、rt 元素与 ruby 元素的使用 41

3.3.3　案例 13——progress 元素的使用 42

3.3.4　案例 14——command 元素的使用 42

3.3.5　案例 15——embed 元素的使用 43

3.3.6 案例 16——details 元素与 summary 元素的使用......43
3.3.7 案例 17——datalist 元素的使用......44
3.4 新增全局属性......45
3.4.1 案例 18——contentEditable 属性的使用......45
3.4.2 案例 19——spellcheck 属性的使用......46
3.4.3 案例 20——tabIndex 属性的使用......46
3.5 新增的其他属性......47
3.5.1 案例 21——表单相关属性的使用......47
3.5.2 案例 22——链接相关属性的使用......54
3.5.3 案例 23——其他新增属性的使用......55
3.6 HTML 5 废除的属性......56
3.7 高手解惑......57

第 II 篇　核 心 技 术

第 4 章　设计网页文本内容......61
4.1 标题文字的建立......62
4.1.1 案例 1——标题文字标记......62
4.1.2 案例 2——标题文字的对齐方式......63
4.2 设置文字格式......63
4.2.1 案例 3——设置文字字体......63
4.2.2 案例 4——设置字号......64
4.2.3 案例 5——设置文字颜色......66
4.2.4 案例 6——设置粗体、斜体、下画线......67
4.2.5 案例 7——设置上标与下标......68
4.2.6 案例 8——设置字体风格......68
4.2.7 案例 9——设置加粗字体......69
4.2.8 案例 10——设置字体复合属性......70
4.2.9 案例 11——设置阴影文本......71
4.2.10 案例 12——控制换行......72
4.3 设置段落格式......73
4.3.1 案例 13——设置段落标记......73
4.3.2 案例 14——设置换行标记......74
4.4 设置网页水平线......75
4.4.1 案例 15——添加水平线......75
4.4.2 案例 16——设置水平线的宽度与高度......75
4.4.3 案例 17——设置水平线的颜色..76
4.4.4 案例 18——设置水平线的对齐方式......76
4.4.5 案例 19——去掉水平线阴影......77
4.5 综合案例——成才教育网文本设计......77
4.6 高手解惑......78
第 5 章　设计网页列表与段落......79
5.1 网页文字列表的设计......80
5.1.1 案例 1——建立无序列表<ul>......80
5.1.2 案例 2——建立有序列表<ol>......81
5.1.3 案例 3——建立不同类型的无序列表......82
5.1.4 案例 4——建立不同类型的有序列表......82
5.1.5 案例 5——嵌套列表......83
5.1.6 案例 6——自定义列表<dl>......83
5.2 网页段落格式的设计......84
5.2.1 案例 7——设计单词间隔 word-spacing......84
5.2.2 案例 8——设计字符间隔 letter-spacing......85
5.2.3 案例 9——设计文字修饰 text-decoration......86
5.2.4 案例 10——设计垂直对齐方式 vertical-align......87
5.2.5 案例 11——设计文本转换 text-transform......88

5.2.6 案例 12——设计水平对齐方式 text-align 89
5.2.7 案例 13——设计文本缩进 text-indent 91
5.2.8 案例 14——设计文本行高 line-height 92
5.2.9 案例 15——处理空白 white-space 93
5.2.10 案例 16——文本反排 unicode-bidi 94
5.3 综合案例——制作图文混排型旅游网页 96
5.4 高手解惑 97

第 6 章 HTML 5 网页中的图像 99

6.1 网页中的图像<img> 100
6.1.1 网页中支持的图片格式 100
6.1.2 图像中的路径 100
6.2 在网页中插入图像 102
6.2.1 案例 1——插入图像 102
6.2.2 案例 2——从不同位置插入图像 103
6.3 编辑网页中的图像 103
6.3.1 案例 3——设置图像的宽度和高度 104
6.3.2 案例 4——设置图像的提示文字 104
6.3.3 案例 5——将图片设置为网页背景 105
6.3.4 案例 6——排列图像 106
6.4 综合案例——图文并茂的房屋装饰装修网页 107
6.5 高手解惑 108

第 7 章 使用 HTML 5 建立超链接 109

7.1 网页超链接的概念 110
7.1.1 什么是网页超链接 110
7.1.2 超链接中的 URL 110
7.1.3 超链接的 URL 类型 111
7.2 建立网页超级链接 111
7.2.1 案例 1——创建超文本链接 111
7.2.2 案例 2——创建图片链接 113
7.2.3 案例 3——创建下载链接 114
7.2.4 案例 4——使用相对路径和绝对路径 115
7.2.5 案例 5——设置以新窗口显示超链接页面 115
7.2.6 案例 6——设置电子邮件链接 116
7.3 案例 7——浮动框架 iframe 的使用 117
7.4 综合案例——使用锚链接制作电子书阅读网页 119
7.5 高手解惑 122

第 8 章 使用 HTML 5 创建表单 123

8.1 案例 1——认识表单 124
8.2 表单基本元素的使用 124
8.2.1 案例 2——单行文本输入框 text 125
8.2.2 案例 3——多行文本输入框 textarea 125
8.2.3 案例 4——密码域 password 126
8.2.4 案例 5——单选按钮 radio 127
8.2.5 案例 6——复选框 checkbox 128
8.2.6 案例 7——列表框 select 128
8.2.7 案例 8——普通按钮 button 129
8.2.8 案例 9——提交按钮 submit 130
8.2.9 案例 10——重置按钮 reset 131
8.3 表单高级元素的使用 132
8.3.1 案例 11——url 属性的使用 132
8.3.2 案例 12——email 属性的使用 132
8.3.3 案例 13——date 和 time 属性的使用 133
8.3.4 案例 14——number 属性的使用 134
8.3.5 案例 15——range 属性的使用 135
8.3.6 案例 16——required 属性的使用 136

8.4 综合案例——创建用户反馈表单 137
8.5 跟我学上机——制作用户注册表单 138
8.6 高手解惑 139

第 9 章 使用 HTML 5 创建表格 141

9.1 案例 1——表格的基本结构 142
9.2 创建表格 143
9.2.1 案例 2——创建普通表格 143
9.2.2 案例 3——创建一个带有标题的表格 144
9.3 编辑表格 145
9.3.1 案例 4——定义表格的边框类型 145
9.3.2 案例 5——定义表格的表头 146
9.3.3 案例 6——设置表格背景 147
9.3.4 案例 7——设置单元格的背景 148
9.3.5 案例 8——合并单元格 148
9.3.6 案例 9——排列单元格中的内容 152
9.3.7 案例 10——设置单元格的行高与列宽 153
9.4 案例 11——完整的表格标记 153
9.5 综合案例——制作计算机报价表 154
9.6 跟我学上机——制作学生成绩表 157
9.7 高手解惑 160

第 10 章 HTML 5 中的音频和视频 161

10.1 audio 标签概述 162
10.1.1 案例 1——认识 audio 标签 162
10.1.2 audio 标签的属性 163
10.1.3 浏览器对 audio 标签的支持情况 163
10.2 在网页中添加音频文件 163
10.2.1 案例 2——添加自动播放的音频文件 163
10.2.2 案例 3——添加带有控件的音频文件 164
10.2.3 案例 4——添加循环播放的音频文件 165
10.2.4 案例 5——添加预播放的音频文件 165
10.3 video 标签概述 166
10.3.1 案例 6——认识 video 标签 166
10.3.2 video 标签的属性 167
10.3.3 浏览器对 video 标签的支持情况 167
10.4 在网页中添加视频文件 168
10.4.1 案例 7——添加自动播放的视频文件 168
10.4.2 案例 8——添加带有控件的视频文件 169
10.4.3 案例 9——添加循环播放的视频文件 169
10.5 综合案例——设置视频文件的高度与宽度 170
10.6 跟我学上机——添加预播放的视频文件 171
10.7 高手解惑 172

第 11 章 使用 HTML 5 绘制图形 173

11.1 添加 canvas 的步骤 174
11.2 绘制基本形状 174
11.2.1 案例 1——绘制矩形 175
11.2.2 案例 2——绘制圆形 176
11.2.3 案例 3——使用 moveTo 与 lineTo 绘制直线 177
11.2.4 案例 4——使用 bezierCurveTo 绘制贝济埃曲线 178
11.3 绘制渐变图形 179
11.3.1 案例 5——绘制线性渐变 179
11.3.2 案例 6——绘制径向渐变 181
11.4 绘制变形图形 182
11.4.1 案例 7——绘制平移效果的图形 182
11.4.2 案例 8——绘制缩放效果的图形 183
11.4.3 案例 9——绘制旋转效果的图形 185

11.4.4 案例 10——绘制组合效果的图形......186
11.4.5 案例 11——绘制带阴影的图形......188
11.5 使用图像......189
11.5.1 案例 12——绘制图像......189
11.5.2 案例 13——平铺图像......190
11.5.3 案例 14——裁剪图像......192
11.5.4 案例 15——图像的像素化处理......193
11.6 案例 16——绘制文字......195
11.7 图形的保存与恢复......196
11.7.1 案例 17——保存与恢复状态......196
11.7.2 案例 18——保存文件......197
11.8 综合案例——绘制火柴棒人物......199
11.9 跟我学上机——绘制商标......201
11.10 高手解惑......203

第 III 篇 高 级 技 能

第 12 章 HTML 5 中的文件与拖放......207
12.1 选择文件......208
12.1.1 案例 1——选择单个文件......208
12.1.2 案例 2——选择多个文件......208
12.2 使用 FileReader 接口读取文件......209
12.2.1 检测浏览器是否支持 FileReader 接口......209
12.2.2 FileReader 接口的方法......210
12.2.3 案例 3——使用 readAsDataURL 方法预览图片......210
12.2.4 案例 4——使用 readAsText 方法读取文本文件......212
12.3 使用 HTML 5 实现文件的拖放......213
12.3.1 认识文件拖放的过程......214
12.3.2 浏览器支持情况......214
12.3.3 案例 5——在网页中拖放图片......215
12.4 综合案例——在网页中来回拖放图片......216
12.5 跟我学上机——在网页中拖放文字......217
12.6 高手解惑......219
第 13 章 定位地理位置技术......221
13.1 Geolocation API 获取地理位置......222
13.1.1 地理地位的原理......222
13.1.2 获取定位信息的方法......222
13.1.3 常用地理定位方法......222
13.1.4 判断浏览器是否支持 HTML 5 获取地理位置信息......223
13.1.5 指定纬度和经度坐标......224
13.1.6 获取当前位置的经度与纬度......225
13.1.7 处理错误和拒绝......227
13.2 目前浏览器对地理定位的支持情况......227
13.3 综合案例——在网页中调用 Google 地图......228
13.4 跟我学上机——持续获取用户移动后的位置......230
13.5 高手解惑......231
第 14 章 Web 存储和通信技术......233
14.1 认识 Web 存储......234
14.1.1 本地存储和 Cookies 的区别......234
14.1.2 Web 存储方法......234
14.2 使用 HTML 5 Web Storage API......234
14.2.1 测试浏览器的支持情况......235
14.2.2 案例 1——使用 sessionStorage 方法创建对象......236
14.2.3 案例 2——使用 localStorage 方法创建对象......237
14.2.4 案例 3——Web Storage API 的其他操作......238
14.2.5 案例 4——使用 JSON 对象存取数据......239

14.3 目前浏览器对 Web 存储的支持情况 241
14.4 跨文档消息传输 242
14.4.1 跨文档消息传输的基本知识 242
14.4.2 案例 5——跨文档通信应用测试 242
14.5 WebSocket API 245
14.5.1 什么是 WebSocket API 245
14.5.2 WebSocket 通信基础 245
14.5.3 案例 6——服务器端使用 Web Socket API 247
14.5.4 案例 7——客户端使用 WebSocket API 250
14.6 综合案例——制作简单 Web 留言本 250
14.7 跟我学上机——编写简单的 WebSocket 服务器 253
14.8 高手解惑 257

第 15 章 处理线程和服务器发送事件 259

15.1 Web Worker 260
15.1.1 Web Worker 概述 260
15.1.2 线程中常用的变量、函数与类 260
15.1.3 案例 1——与线程进行数据的交互 261
15.2 线程嵌套 263
15.2.1 案例 2——单线程嵌套 263
15.2.2 案例 3——多个子线程中的数据交互 265
15.3 服务器发送事件概述 267
15.4 服务器发送事件的实现过程 267
15.4.1 案例 4——检测浏览器是否支持 Server-Sent 事件 267
15.4.2 案例 5——使用 EventSource 对象 268
15.4.3 案例 6——编写服务器端代码 268
15.5 综合案例——创建 Web Worker 计数器 269
15.6 跟我学上机——服务器发送事件实战应用 270
15.7 高手解惑 272

第 16 章 构建离线的 Web 应用 273

16.1 HTML 5 离线 Web 应用概述 274
16.2 案例 1——使用 HTML 5 离线 Web 应用 API 274
16.2.1 检查浏览器的支持情况 274
16.2.2 搭建简单的离线应用程序 275
16.2.3 支持离线行为 275
16.2.4 Manifest 文件 276
16.2.5 Application Cache API 277
16.3 案例 2——使用 HTML 5 离线 Web 应用构建应用 278
16.3.1 创建记录资源的 manifest 文件 278
16.3.2 创建构成界面的 HTML 和 CSS 279
16.3.3 创建离线的 JavaScript 279
16.3.4 检查 Application Cache 的支持情况 281
16.3.5 为 Update 按钮添加处理函数 281
16.3.6 添加 Storage 功能代码 282
16.3.7 添加离线事件处理程序 282
16.4 综合案例——离线定位跟踪 283
16.5 高手解惑 287

第 IV 篇 移 动 开 发

第 17 章 jQuery Mobile 基础 291

17.1 认识 jQuery Mobile 292
17.2 跨平台移动设备网页 jQuery Mobile 292

17.2.1 案例 1——移动设备模拟器 292
17.2.2 案例 2——jQuery Mobile 的安装 294
17.2.3 案例 3—— jQuery Mobile 网页的架构 296
17.3 案例 4——创建多页面的 jQuery Mobile 网页 297
17.4 案例 5——将页面作为对话框使用 298
17.5 案例 6——绚丽多彩的页面切换效果 299
17.6 高手解惑 301

第 18 章 jQuery Mobile UI 组件 303

18.1 套用 UI 组件 304
18.1.1 案例 1——表单组件 304
18.1.2 案例 2——按钮和组按钮 311
18.1.3 案例 3——按钮图标 313
18.1.4 案例 4——弹窗 315
18.2 列表 316
18.2.1 案例 5——列表视图 316
18.2.2 案例 6——列表内容 319
18.2.3 案例 7——列表过滤 320
18.3 面板和可折叠块 322
18.3.1 案例 8——面板 322
18.3.2 案例 9——可折叠块 323
18.4 案例 10——导航条 325
18.5 综合案例——使用 jQuery Mobile 主题 328
18.6 高手解惑 331

第 19 章 jQuery Mobile 事件 333

19.1 页面事件 334
19.1.1 案例 1——初始化事件 334
19.1.2 案例 2——外部页面加载事件 336
19.1.3 案例 3——页面过渡事件 338
19.2 触摸事件 340
19.2.1 案例 4——点击事件 340
19.2.2 案例 5——滑动事件 342
19.3 案例 6——滚屏事件 344
19.4 案例 7——定位事件 347
19.5 高手解惑 349

第 20 章 数据存储和读取技术 351

20.1 Web SQL Database 概述 352
20.2 数据库的基本操作 352
20.3 数据表的基本操作 353
20.4 数据的基本操作 355
20.5 综合案例——Web SQL Database 的综合操作技能 356
20.6 高手解惑 359

第 V 篇 综合案例实战

第 21 章 制作休闲娱乐类网页 363

21.1 整体布局 364
21.1.1 设计分析 364
21.1.2 排版架构 365
21.2 模块组成 365
21.3 制作步骤 366
21.3.1 制作样式表 366
21.3.2 Logo 与导航菜单 373
21.3.3 搜索条 374
21.3.4 左侧视频模块 374
21.3.5 评论模块 375
21.3.6 右侧热门推荐 377
21.3.7 底部模块 378

第 22 章 制作企业门户类网页 381

22.1 整体布局 382
22.1.1 设计分析 382
22.1.2 排版架构 383
22.2 模块组成 383
22.3 制作步骤 384
22.3.1 样式表 384

22.3.2 网页头部......385
22.3.3 导航菜单栏......386
22.3.4 中间主体第一栏......386
22.3.5 中间主体第二栏......390
22.3.6 中间主体第三栏......393
22.3.7 中间主体第四栏......398
22.3.8 中间主体第五栏......402
22.3.9 网页底部......407

第 23 章 制作电子商务类网页......409

23.1 整体布局......410
23.1.1 设计分析......410
23.1.2 排版架构......411
23.2 模块组成......411
23.3 制作步骤......411
23.3.1 样式表......411
23.3.2 网页头部......419
23.3.3 主体第一通栏......420
23.3.4 主体第二通栏......420
23.3.5 主体第三通栏......421
23.3.6 网页底部......422

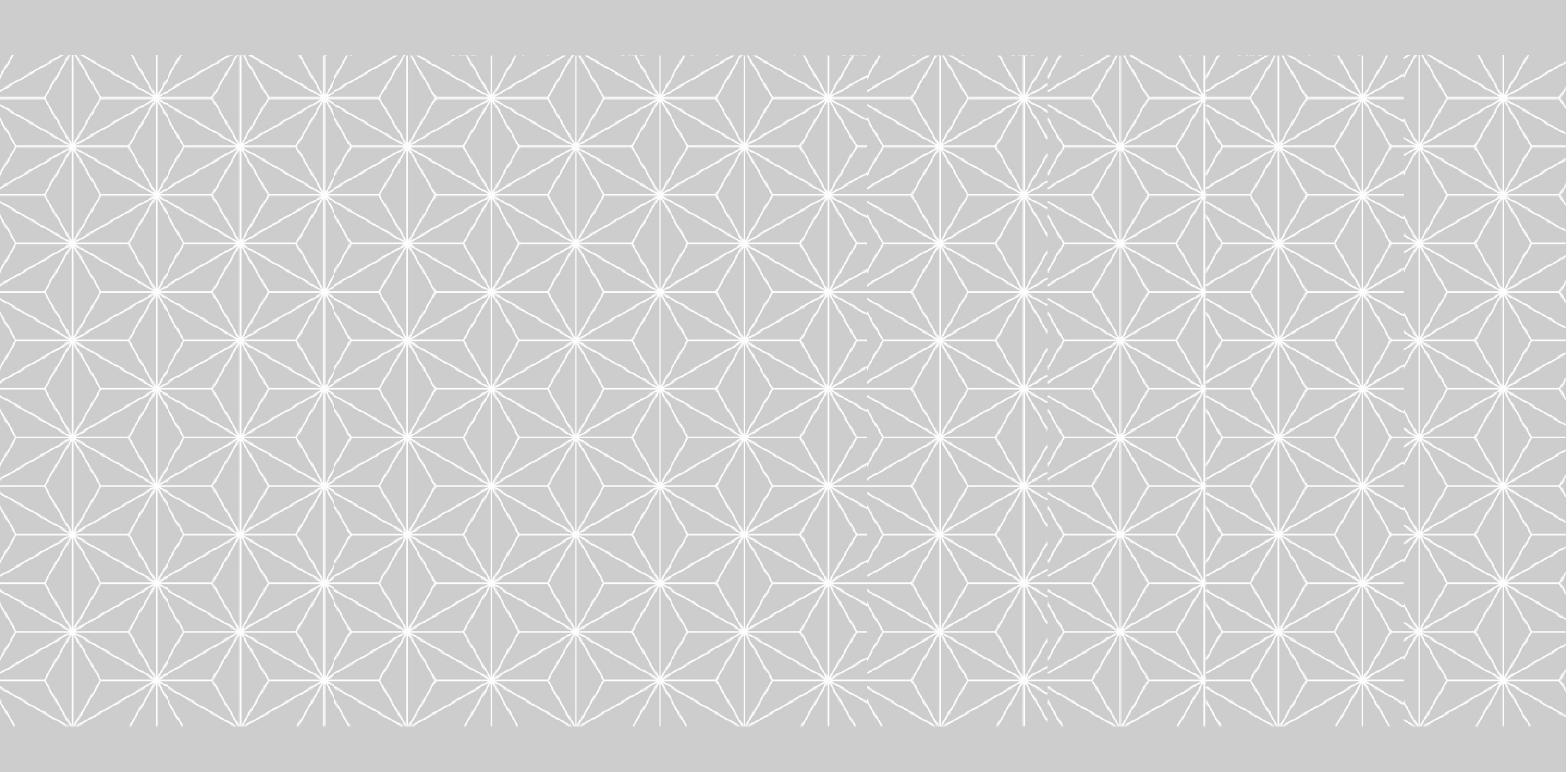
第 24 章 开发连锁酒店订购系统......425

24.1 连锁酒店订购的需求分析......426
24.2 网站的结构......426
24.3 连锁酒店订购系统的代码实现......427
24.3.1 设计首页......427
24.3.2 订购页面......428
24.3.3 连锁分店页面......433
24.3.4 查看订单页面......434
24.3.5 酒店介绍页面......436

第 I 篇

基 础 入 门

- 第 1 章　新一代 Web 前端技术 HTML 5
- 第 2 章　HTML 5 网页的文档结构
- 第 3 章　HTML 5 与 HTML 4 的区别





第 1 章 新一代 Web 前端技术 HTML 5

目前，网络已经成为人们娱乐、工作中不可或缺的一部分。网页设计也成为学习计算机知识的重要内容之一。制作网页可以采用可视化编辑软件，但是无论采用哪一种网页编辑软件，最后都是将所设计的网页转化为 HTML。

HTML 是网页设计的基础语言。本章将介绍 HTML 的基本概念和编写方法及浏览 HTML 文件的方法，使读者初步了解 HTML，为后面的学习打下基础。

重点案例效果



1.1 HTML 的基本概念

互联网上的信息是以网页形式展示给用户的，网页是网络信息传递的载体。网页文件是用标记语言书写的，这种语言称为超文本标记语言(Hyper Text Markup Language，HTML)。

1.1.1 HTML 的发展历程

HTML 是一种描述语言，而不是一种编程语言，主要用于描述超文本中内容的显示方式。标记语言从诞生至今，经历了 20 多年，发展过程中也有很多曲折，经历的版本及发布日期如表 1-1 所示。

表 1-1 超文本标记语言的发展历程

版　本	发布日期	说　明
超文本标记语言(第 1 版)	1993 年 6 月	作为互联网工程工作小组(IETF)工作草案发布(并非标准)
HTML 2.0	1995 年 11 月	作为 RFC 1866 发布，在 RFC 2854 于 2000 年 6 月发布之后被宣布已经过时
HTML 3.2	1996 年 1 月 14 日	W3C 推荐标准
HTML 4.0	1997 年 12 月 18 日	W3C 推荐标准
HTML 4.01	1999 年 12 月 24 日	微小改进，W3C 推荐标准
ISO HTML	2000 年 5 月 15 日	基于严格的 HTML 4.01 语法，是国际标准化组织和国际电工委员会的标准
XHTML 1.0	2000 年 1 月 26 日	W3C 推荐标准(修订后于 2002 年 8 月 1 日重新发布)
XHTML 1.1	2001 年 5 月 31 日	较 1.0 有微小改进
XHTML 2.0 草案	没有发布	2009 年，W3C 停止了 XHTML 2.0 工作组的工作
HTML 5	2014 年 10 月	HTML 5 标准规范最终制定完成

1.1.2 什么是 HTML

HTML 5 不是一种编程语言，而是一种描述性的标记语言，用于描述超文本中的内容和结构。HTML 最基本的语法是<标记符></标记符>。标记符通常都是成对使用的，有一个开头标记和一个结束标记。结束标记只是在开头标记的前面加一个斜杠“/”。当浏览器收到 HTML 文件后，就会解释里面的标记符，然后把标记符相对应的功能表达出来。

例如，在 HTML 中用<p></p>标记符来定义一个换行符。当浏览器遇到<p></p>标记符时，会把该标记中的内容自动形成一个段落。当遇到
标记符时，会自动换行，并且该标记符后的内容会从一个新行开始。这里的
标记符是单标记，没有结束标记，标记后的“/”符号可以省略；但为了使代码规范，一般建议加上。

1.1.3 HTML 5 文件的基本结构

完整的 HTML 文件包括标题、段落、列表、表格、绘制的图形及各种嵌入对象，这些对象统称为 HTML 元素。一个 HTML 5 文件的基本结构如下：

```
<!DOCTYPE html>
<html>                  文件开始的标记
<head>                  文件头部开始的标记
...                     文件头部的内容
</head>                 文件头部结束的标记
<body>                  文件主体开始的标记
...                     文件主体的内容
</body>                 文件主体结束的标记
</html>                 文件结束的标记
```

从上述代码结构可以看出，在 HTML 文件中，所有的标记都是相对应的，开头标记为< >，结束标记为</>，在这两个标记中间添加内容。这些基本标记的使用方法及详细解释见第 2 章。

1.2 HTML 5 的优势

从 HTML 4.0、XHTML 到 HTML 5，在某种意义上讲，这是 HTML 描述性标记语言的一种更加规范的过程。因此，HTML 5 并没有给开发者带来多大的冲击。但 HTML 5 也增加了很多非常实用的新功能。下面来介绍 HTML 5 的一些优势。

1.2.1 解决了跨浏览器问题

浏览器是网页的运行环境，因此浏览器的类型也是在网页设计时会遇到的一个问题。由于各个软件厂商对 HTML 标准的支持有所不同，导致同样的网页在不同的浏览器中会有不同的表现。并且 HTML 5 新增的功能在各个浏览器中的支持程度也不一致，浏览器的因素变得比以往传统的网页设计更加重要。

为了保证设计出来的网页在不同浏览器中的效果一致，HTML 5 会让问题简单化，具备友好的跨浏览器性能。针对不支持新标签的老式 IE 浏览器，用户只要简单地添加 JavaScript 代码，就可以让它们使用新的 HTML 5 元素。

1.2.2 新增了多个新特性

HTML 语言从 1.0 至 5.0 经历了巨大变化，从单一的文本显示功能到图文并茂的多媒体显示功能，许多特性经过多年的完善，已经成为一种非常重要的标记语言。尤其是 HTML 5，对多媒体的支持功能更强，它具备如下功能。

(1) 新增了语义化标签，使文档结构明确。

(2) 新的文档对象模型(DOM)。

(3) 实现了 2D 绘图的 Canvas 对象。
(4) 可控媒体播放。
(5) 离线存储。
(6) 文档编辑。
(7) 拖放。
(8) 跨文档消息。
(9) 浏览器历史管理。
(10) MIME 类型和协议注册。

对于这些新功能，支持 HTML 5 的浏览器在处理 HTML 5 代码错误的时候必须更灵活，而那些不支持 HTML 5 的浏览器将忽略 HTML 5 代码。

1.2.3 用户优先的原则

HTML 5 标准的制定是以用户优先为原则的，一旦遇到无法解决的冲突时，规范会把用户放到第一位，其次是网页的作者，再次是浏览器，接着是规范制定者(W3C/WHATWG)，最后才考虑理论的纯粹性。所以，总体来看，HTML 5 的绝大部分特性还是实用的，只是有些情况下还不够完美。

举例说明一下，下述三行代码虽然有所不同，但在 HTML 5 中都能被正确识别：

```
id="HTML 5"
id=HTML 5
ID="HTML 5"
```

在上述示例中，除了第一个外，另外两个语法都不是很严格，这种不严格的语法被广泛使用，受到一些技术人员的反对。但是无论语法严格与否，对网页查看者来说都没有任何影响，他们只需要看到想要的网页效果就可以了。

为了增强 HTML 5 的使用体验，还加强了以下两个方面的设计。

1. 安全机制的设计

为确保 HTML 5 的安全，在设计 HTML 5 时做了很多针对安全的设计。HTML 5 引入了一种新的基于来源的安全模型，该模型不仅易用，而且对各种不同的 API 都通用。使用这个安全模型，可以做一些以前做不到的事情，不需要借助于任何所谓聪明、有创意却不安全的 Hack 就能跨域进行安全对话。

2. 表现和内容分离

表现和内容分离是 HTML 5 设计中的另一个重要内容。HTML 5 在所有可能的地方都努力进行了分离，也包括 CSS。实际上，表现和内容的分离早在 HTML 4 中就有设计，只是分离得并不彻底。为了避免可访问性差、代码高复杂度、文件过大等问题，HTML 5 规范中更细致、清晰地分离了表现和内容。但是出于 HTML 5 兼容性的考虑，一些老的表现和内容的代码还是可以兼容使用的。

1.2.4 化繁为简的优势

作为当下流行的通用标记语言，HTML 5 越简单越好。所以在设计 HTML 5 时，严格遵循了“简单至上”的原则，主要体现在以下几个方面。

(1) 新的简化的字符集声明。

(2) 新的简化的 DOCTYPE。

(3) 简单而强大的 HTML 5 API。

(4) 以浏览器原生能力替代复杂的 JavaScript 代码。

为了实现以上这些简化操作，HTML 5 规范需要比以前更加细致、精确，比以往任何版本的 HTML 规范都要精确。

在 HTML 5 规范细化的过程中，为了避免造成误解，几乎对所有内容都给出了彻底、完全的定义，特别是对 Web 应用。

基于多种改进过的、强大的错误处理方案，HTML 5 具备了良好的错误处理机制。具体来讲，HTML 5 提倡重大错误的平缓恢复，再次把最终用户的利益放在了第一位。举例说，如果页面中有错误的话，在以前，可能会影响整个页面的显示，而在 HTML 5 中，不会出现这种情况，取而代之的是以标准方式显示 broken 标记，这要归功于 HTML 5 中精确定义的错误恢复机制。

1.3 HTML 5 网页的开发环境

有两种方式可以产生 HTML 文件：一是自己写 HTML 文件，事实上这并不是很困难，也不需要特别的技巧；二是使用 HTML 编辑器，它可以辅助使用者来做编写工作。

1.3.1 案例 1——使用记事本手工编写 HTML 5

前面介绍过，HTML 5 是一种标记语言，其代码是以文本形式存在的。因此，所有的记事本工具都可以作为它的开发环境。

HTML 文件的扩展名为.html 或.htm，将 HTML 源代码输入到记事本并保存之后，可以在浏览器中打开文档以查看其效果。

使用记事本编写 HTML 文件的具体操作步骤如下。

step 01 单击 Windows 桌面上的“开始”按钮，选择“所有程序”→“附件”→“记事本”命令，打开一个记事本，在记事本中输入 HTML 代码，如图 1-1 所示。

step 02 编辑完 HTML 文件后，选择“文件”→“保存”菜单命令或按 Ctrl+S 组合键，在弹出的“另存为”对话框中，设置“保存类型”为“所有文件”，然后将文件扩展名设为.html 或.htm，如图 1-2 所示。

step 03 单击“保存”按钮，即可保存文件。打开网页文档，在浏览器中预览，如图 1-3 所示。

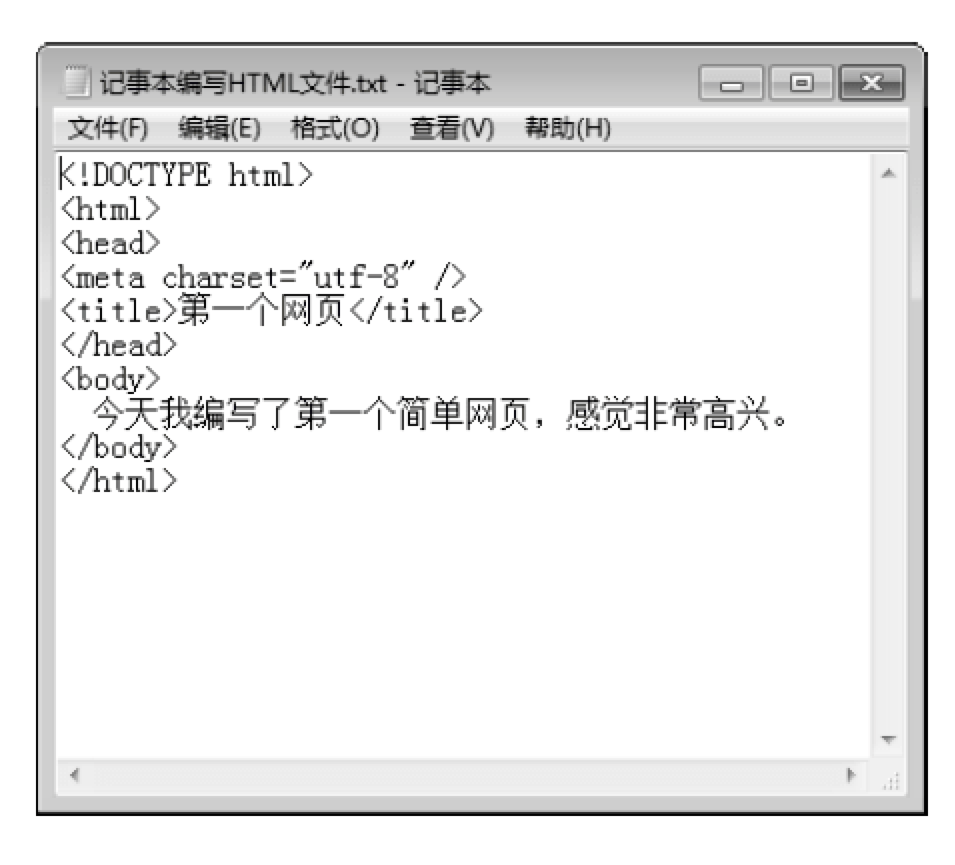



图 1-1　输入 HTML 代码



图 1-2　“另存为”对话框



图 1-3　网页的浏览效果

1.3.2　案例 2——使用 Dreamweaver CC 编写 HTML 文件

“工欲善其事，必先利其器”，虽然使用记事本可以编写 HTML 文件，但是编写效率太低，对于语法错误及格式都没有提示。于是，很多专门编写 HTML 网页的工具弥补了这种缺陷。Adobe 公司的 Dreamweaver CC 用户界面非常友好，是一款非常优秀的网页开发工具，深受广大用户的喜爱。Dreamweaver CC 的主界面如图 1-4 所示。

1. 文档窗口

文档窗口位于界面的中部，它是用来编排网页的区域，与在浏览器中的结果相似。在文档窗口中，可以将文档分为 3 种视图显示模式。

(1) 代码视图。使用代码视图，可以在文档窗口中显示当前文档的源代码，也可以在该窗口中直接输入 HTML 代码。

(2) 设计视图。在设计视图下无须编辑任何代码，可以直接使用可视化的操作编辑网页。

(3) 拆分视图。在拆分视图下，左半部分显示代码视图，右半部分显示设计视图。在这

种视图模式下，可以通过输入 HTML 代码直接观看效果，还可以通过设计视图插入对象，直接查看源文件。

在各种视图间进行切换时，只需要在文档工具栏中单击相应的视图按钮即可。文档工具栏如图 1-5 所示。



图 1-4 Dreamweaver CC 的主界面

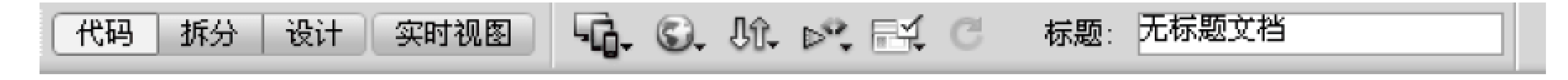


图 1-5 文档工具栏

2. “插入”面板

“插入”面板是在设计视图下使用频度很高的面板之一。“插入”面板默认打开的是“常用”页，它包括了最常用的一些对象。例如，在文档中的光标位置插入一段文本、图像或表格等。用户可以根据需要切换到其他页，如图 1-6 所示。



图 1-6 “插入”面板

3. “属性”面板

“属性”面板中主要包含当前选择的对象相关属性的设置。可以通过菜单栏中的“窗口”→“属性”命令或 Ctrl+F3 组合键打开或关闭“属性”面板。

“属性”面板是常用的一个面板，因为无论要编辑哪个对象的属性，都要用到它。其内容也会随着选择对象的不同而改变。例如，当光标定位在文档主体文字内容部分时，“属性”面板显示文字的相关属性，如图 1-7 所示。

Dreamweaver CC 中还有很多面板，在以后使用时，再做详细讲解。打开的面板越多，编辑文档的区域就会越小，为了便于编辑文档，可以通过 F4 功能键快速隐藏和显示所有面板。



图 1-7　“属性”面板

使用 Dreamweaver CC 编写 HTML 文件的具体操作步骤如下。

step 01 启动 Dreamweaver CC，如图 1-8 所示，在欢迎屏幕的“新建”栏中选择 HTML 选项。或者选择菜单栏中的“文件”→“新建”命令(快捷键为 Ctrl+N)。

step 02 弹出“新建文档”对话框，如图 1-9 所示，在“页面类型”列表框中选择“HTML 模板”选项。

step 03 单击“创建”按钮，创建 HTML 文件，如图 1-10 所示。



图 1-8　包含欢迎屏幕的主界面

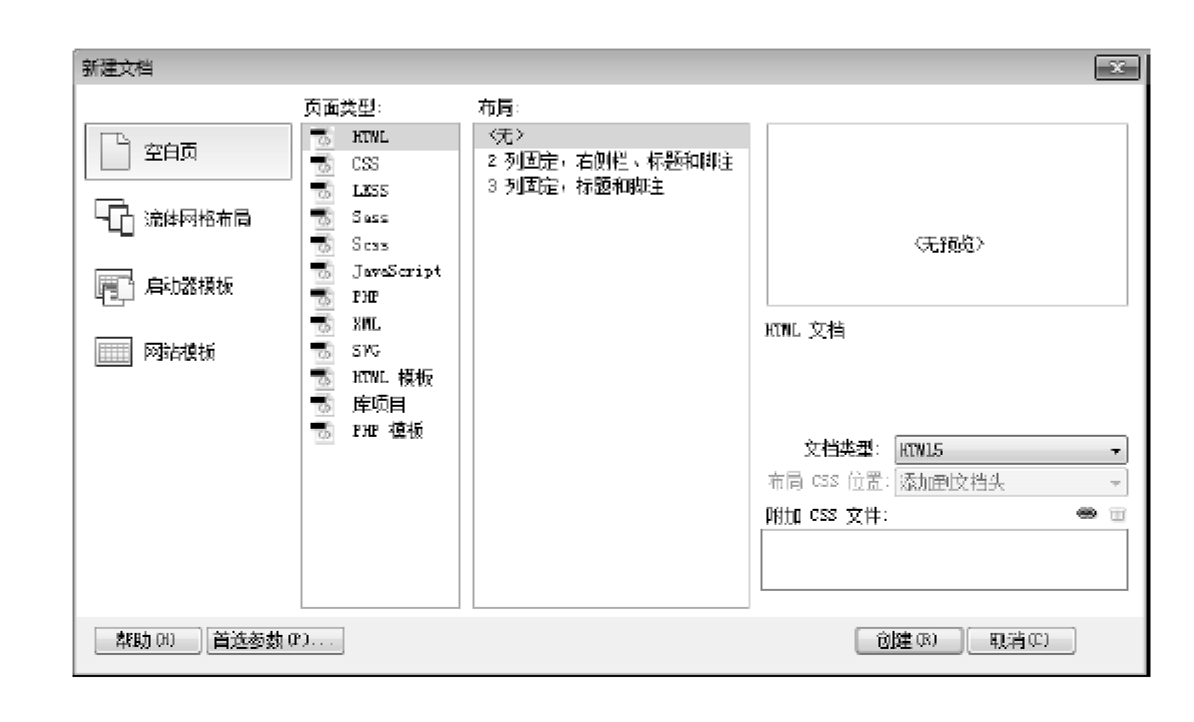


图 1-9　“新建文档”对话框

step 04 在文档工具栏中，单击“代码”按钮，切换到代码视图，如图 1-11 所示。

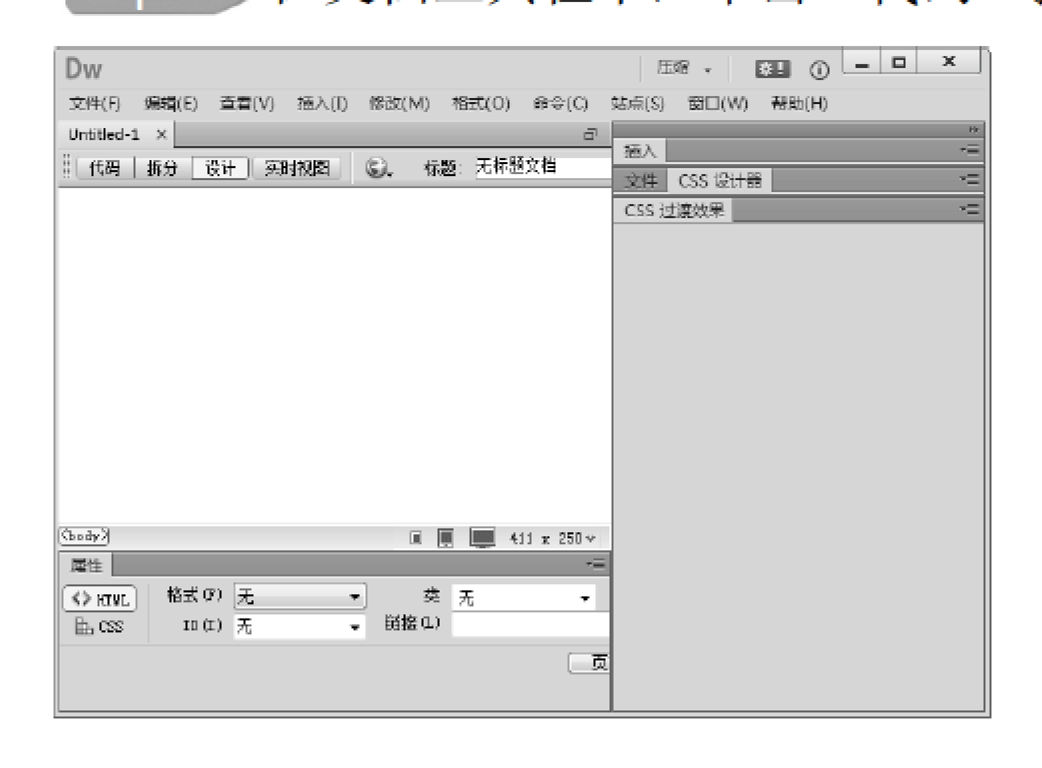


图 1-10　在设计视图下显示创建的文档

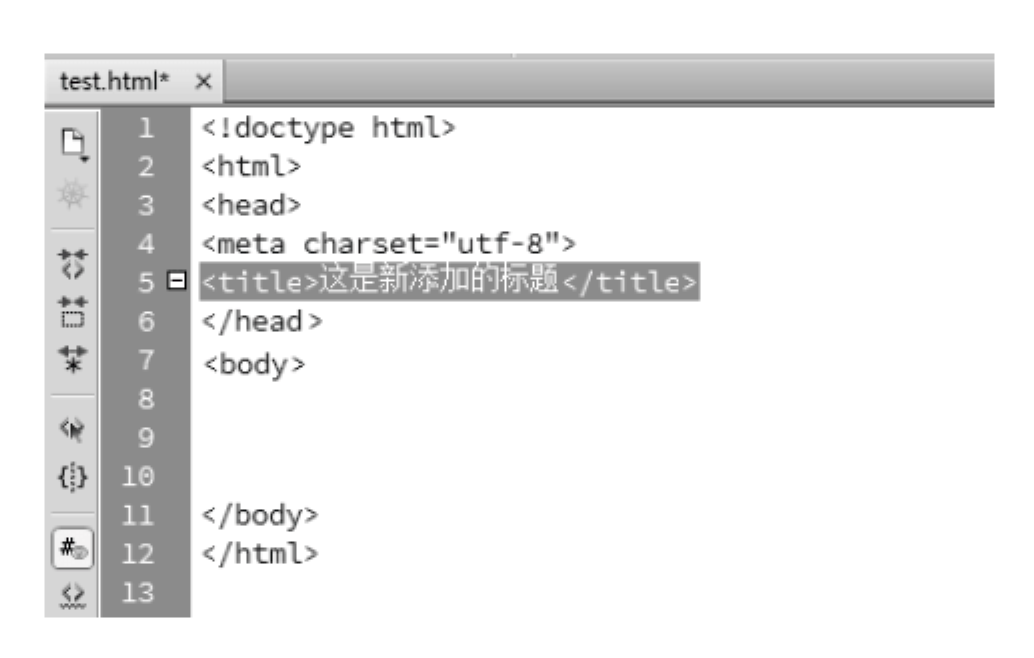


图 1-11　在代码视图下显示创建的文档

step 05 修改 HTML 文档标题，将代码中<title>标记中的网页标题修改成“第一个网页”。然后在<body>标记中输入“今天我使用 Dreamweaver CC 编写了第一个简单网页，感觉非常高兴。”。

完整的 HTML 代码如下所示：

```
<!doctype html >
<html>
```

```
<head>
<meta charset=utf-8" />
<title>第一个网页</title>
</head>
<body>
今天我使用 Dreamweaver CC 编写了第一个简单网页，感觉非常高兴。
</body>
</html>
```

step 06 保存文件。从菜单栏中选择“文件”→“保存”命令或按 Ctrl+S 组合键，弹出“另存为”对话框。在该对话框中设置保存位置，并输入文件名，单击“保存”按钮，如图 1-12 所示。

step 07 单击文档工具栏中的图标，选择查看网页的浏览器，或按功能键 F12，使用默认浏览器查看网页，预览效果如图 1-13 所示。




图 1-12 保存文件



图 1-13 浏览器预览效果

1.4 使用浏览器查看 HTML 5 文件

开发者经常需要查看 HTML 源代码及其效果。使用浏览器可以查看网页的显示效果，也可以在浏览器中直接查看 HTML 源代码。

1.4.1 案例 3——查看页面效果

前面已经介绍过，为了测试网页的兼容性，可以在不同的浏览器中打开网页。在非默认浏览器中打开网页的方法有很多种，在此介绍两种常用的方法。

(1) 选择浏览器中的“文件”→“打开”命令(有些浏览器的菜单命令为“打开文件”)，选择要打开的网页即可，如图 1-14 所示。

(2) 在 HTML 文件上右击，从弹出的快捷菜单中选择“打开方式”命令，然后选择需要的浏览器，如图 1-15 所示。如果浏览器没有出现在菜单中，可以选择“选择其他应用”命令，在计算机中查找浏览器程序。

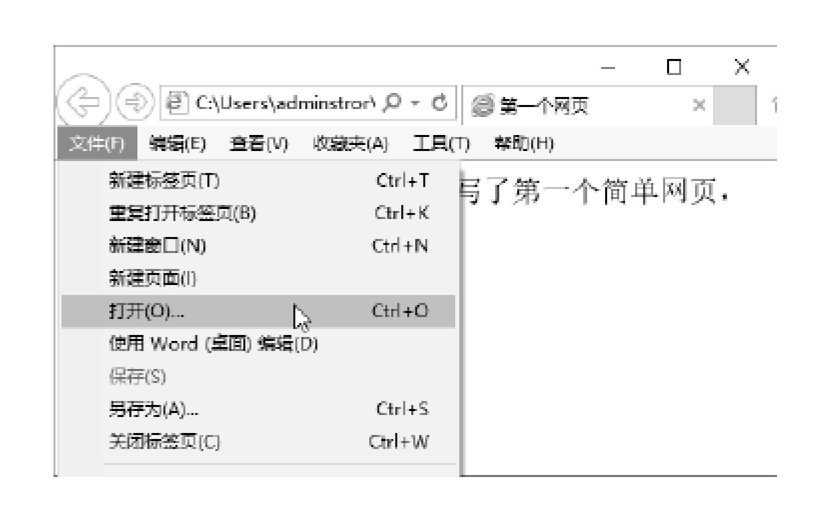


图 1-14　选择“打开”命令

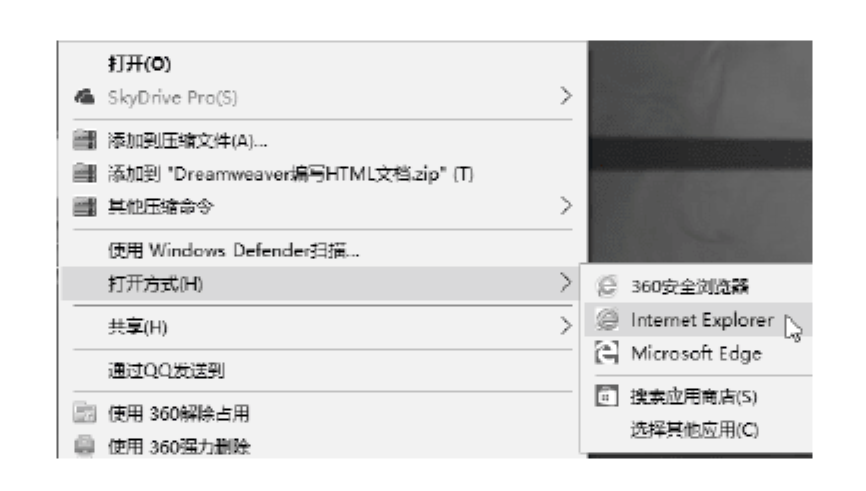


图 1-15　选择不同的浏览器来打开网页

1.4.2　案例 4——查看源文件

查看网页源代码的常见方法有以下两种。

(1)　在页面空白处右击，从弹出的快捷菜单中选择“查看源”命令，如图 1-16 所示。

(2)　在浏览器的菜单栏中选择“查看”→“源”命令，如图 1-17 所示。



图 1-16　选择“查看源”命令



图 1-17　选择“源”命令



由于浏览器的规定各不相同，有些浏览器将“查看源”命名为“查看源代码”，但是，操作方法完全相同。

1.5　高 手 解 惑

疑问 1：为何使用记事本编辑的 HTML 文件无法在浏览器中预览，而是直接在记事本中打开？

答：很多初学者，在保存文件时，没有将 HTML 文件的扩展名.html 或.htm 作为文件的后缀，导致文件还是以.txt 为扩展名，因此无法在浏览器中查看。如果读者是通过右击鼠标创建记事本文件的，在为文件重命名时，一定要以.html 或.htm 作为文件的后缀。特别要注意的是，当 Windows 系统的扩展名隐藏时，更容易出现这样的错误。读者可以在“文件夹选项”对话框中查看是否显示扩展名。

疑问 2：如何显示或隐藏 Dreamweaver CC 的欢迎屏幕？

答：Dreamweaver CC 的欢迎屏幕可以帮助使用者快速进行打开文件、新建文件和获取相关帮助的操作。如果读者不希望显示该欢迎屏幕，可以按 Ctrl+U 组合键，在弹出的窗口中，选择左侧的“常规”页，将右侧“文档选项”部分的“显示欢迎屏幕”复选框取消勾选。

第 2 章

HTML 5 网页的文档结构

一个完整的 HTML 5 网页文档包括标题、段落、列表、表格、绘制的图形及各种嵌入对象，这些对象统称为 HTML 5 元素。本章详细介绍 HTML 5 网页文档的基本结构。

重点案例效果



2.1 HTML 5 文件的基本结构

在一个 HTML 5 文档中，必须包含<HTML></HTML>标记，并且放在一个 HTML 5 文档中的开始和结束位置。即每个文档以<HTML>开始，以</HTML>结束。<HTML></HTML>之间通常包含两个部分，分别为<HEAD></HEAD>和<BODY></BODY>。HEAD 标记包含 HTML 头部信息，如文档标题、样式定义等。BODY 包含文档主体部分，即网页内容。需要注意的是，HTML 标记不区分大小写。

2.1.1 HTML 5 页面的整体结构

为了便于读者从整体上把握 HTML 5 的文档结构，我们通过一个 HTML 5 页面来介绍 HTML 5 页面的整体结构。示例代码如下：

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
    <TITLE>网页标题</TITLE>
</HEAD>
<BODY>
    网页内容
</BODY>
</HTML>
```

从上述代码可以看出，一个基本的 HTML 5 网页由以下几个部分构成。

(1) <!DOCTYPE HTML>声明。该声明必须位于 HTML 5 文档中的第一行，也就是位于<HTML>标记之前。该标记告知浏览器文档所使用的 HTML 规范。<!DOCTYPE HTML>声明不属于 HTML 标记；它是一条指令，告诉浏览器编写页面所用的标记的版本。由于 HTML 5 版本还没有得到浏览器的完全认可，后面介绍时还采用以前的通用标准。

(2) <HTML></HTML>标记。说明本页面是用 HTML 语言编写的，使浏览器软件能够准确无误地解释和显示。

(3) <HEAD></HEAD>标记。这是 HTML 的头部标记。头部信息不显示在网页中。此标记内可以包含一些其他标记，用于说明文件标题和整个文件的一些公用属性。可以通过<style>标记定义 CSS 样式表，通过<script>标记定义 JavaScript 脚本文件。

(4) <TITLE></TITLE>标记。TITLE 是 HEAD 中的重要组成部分，它包含的内容显示在浏览器的窗口标题栏中。如果没有 TITLE，浏览器标题栏将显示本页的文件名。

(5) <BODY></BODY>标记。BODY 包含 HTML 页面的实际内容，显示在浏览器窗口的客户区中。例如，在页面中，文字、图像、动画、超链接以及其他 HTML 相关的内容都是定义在 BODY 标记里面的。

2.1.2 HTML 5 新增的结构标记

HTML 5 新增的结构标记有<footer></footer>和<header></header>标记。但是，这两个标

记还没有获得大多数浏览器的支持，这里只简单地介绍一下。

<header>标记定义文档的页眉(介绍信息)。使用示例如下：

```
<header>
<h1>欢迎访问主页</h1>
</header>
```

<footer>标记定义 section 或 document 的页脚。在典型情况下，该元素会包含创作者的姓名、文档的创作日期或者联系信息。使用示例如下：

```
<footer>作者：元澈  联系方式：13012345678</footer>
```

2.2 HTML 5 基本标记详解

HTML 文档最基本的结构主要包括文档类型说明、HTML 文档开始标记、元信息、主体标记和页面注释标记。

2.2.1 文档类型说明

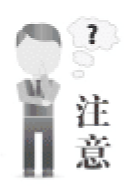
基于 HTML 5 设计准则中的化繁为简原则，Web 页面的文档类型说明(DOCTYPE)被极大地简化了。

细心的读者会发现，在第 1 章中使用 Dreamweaver CC 创建 HTML 5 文档时，文档头部的类型说明代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

上述代码为 XHTML 文档类型说明。可以看到，这段代码既麻烦又难记。HTML 5 对文档类型进行了简化，简单到 15 个字符即可，代码如下：

```
<!DOCTYPE html>
```



DOCTYPE 声明需要出现在 HTML 5 文件的第一行。

2.2.2 HTML 标记

HTML 标记代表文档的开始。由于 HTML 5 语言语法的松散特性，该标记可以省略。但是为了使之符合 Web 标准和体现文档的完整性，应养成良好的编写习惯，这里建议不要省略该标记。

HTML 标记以<html>开头，以</html>结尾，文档的所有内容书写在开头和结尾的中间部分。其语法格式如下：

```
<html>
...
</html>
```

2.2.3 头标记 head

头标记 head 用于说明文档头部的相关信息，一般包括标题信息、元信息、定义 CSS 样式和脚本代码等。HTML 的头部信息以<head>开始，以</head>结束，语法格式如下：

```
<head>
...
</head>
```



<head>元素的作用范围是整篇文档，并且定义在 HTML 语言头部的内容往往不会在网页上直接显示。

在头标记<head>与</head>之间还可以插入标题标记 title 和元信息标记 meta 等。

1．标题标记 title

HTML 页面的标题一般是用来说明页面用途的，它显示在浏览器的标题栏中。在 HTML 文档中，标题信息设置在<head>与</head>之间。标题标记以<title>开始，以</title>结束，语法格式如下：

```
<title>
...
</title>
```

在标记中间的“…”就是标题的内容，它可以帮助用户更好地识别页面。在预览网页时，设置的标题在浏览器的左上方标题栏中显示，如图 2-1 所示。此外，在 Windows 任务栏中显示的也是这个标题。页面的标题只有一个，位于 HTML 文档的头部。



图 2-1　标题栏在浏览器中的显示效果

2．元信息标记 meta

<meta>元素可提供有关页面的元信息(meta-information)，比如针对搜索引擎和更新频度的描述和关键词。<meta>标记位于文档的头部，不包含任何内容。<meta>标记的属性定义了与文档相关联的名称/值对。<meta>标记提供的属性及其取值如表 2-1 所示。

表 2-1 <meta>标记提供的属性及其取值

属　性	值	描　述
charset	character encoding	定义文档的字符编码
content	some_text	定义与 http-equiv 或 name 属性相关的元信息
http-equiv	content-type expires refresh set-cookie	把 content 属性关联到 HTTP 头部
name	author description keywords generator revised others	把 content 属性关联到一个名称

1) 字符集 charset 属性

在 HTML 5 中，有一个新的 charset 属性，它使字符集的定义更加容易。例如，下面的代码告诉浏览器，网页使用 ISO-8859-1 字符集显示：

```
<meta charset="ISO-8859-1">
```

2) 搜索引擎的关键词

在早期，meta keywords 关键词对搜索引擎的排名算法起到一定的作用，也是很多人进行网页优化的基础。关键词在浏览时是看不到的，使用格式如下：

```
<meta name="keywords" content="关键词,keywords" />
```



(1) 不同的关键词之间应使用半角逗号隔开(英文输入状态下)，不要使用"空格"或"|"间隔。

(2) 是 keywords，不是 keyword。

(3) 关键词标签中的内容应该是一个个短语，而不是一段话。

例如，定义针对搜索引擎的关键词，代码如下：

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript" />
```

关键词标签 keywords，曾经是搜索引擎排名中很重要的因素，但现在已经被很多搜索引擎完全忽略。如果我们加上这个标签，对网页的综合表现没有坏处。不过，如果使用不恰当的话，对网页非但没有好处，还有欺诈的嫌疑。在使用关键词标签 keywords 时，要注意以下几点。

(1) 关键词标签中的内容要与网页核心内容相关，应当确信使用的关键词出现在网页文本中。

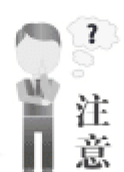
(2) 应当使用用户易于通过搜索引擎检索的关键词，过于生僻的词汇不太适合作为 meta

标签中的关键词。

(3) 不要重复使用关键词，否则可能会被搜索引擎惩罚。

(4) 一个网页的关键词标签里最多包含 3～5 个最重要的关键词，不要超过 5 个。

(5) 每个网页的关键词应该不一样。



由于设计者或 SEO 优化者以前对 meta keywords 关键词的滥用，导致目前它在搜索引擎排名中的作用很小。

3) 页面描述

meta description 元标签(描述元标签)是一种 HTML 元标签，用来简略描述网页的主要内容，通常是被搜索引擎用在搜索结果页上展示给最终用户看的一段文字。页面描述在网页中并不显示出来。页面描述的使用格式如下：

```
<meta name="description" content="网页的介绍" />
```

例如，定义对页面的描述，代码如下：

```
<meta name="description" content="免费的 Web 技术教程。" />
```

4) 页面定时跳转

使用<meta>标记可以使网页在经过一定时间后自动刷新，这可以通过将 http-equiv 属性值设置为 refresh 来实现。content 属性值可以设置为更新时间。

在浏览网页时经常会看到一些欢迎信息的页面，在经过一段时间后，这些页面会自动转到其他页面，这就是网页的跳转。页面定时刷新跳转的语法格式如下：

```
<meta http-equiv="refresh" content="秒;[url=网址]" />
```



上面的[url=网址]部分是可选项，如果有这部分，页面定时刷新并跳转；如果省略该部分，页面只定时刷新，不进行跳转。

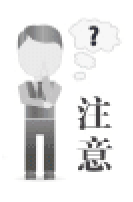
例如，实现每 5 秒刷新一次页面。将下述代码放入 head 标记中即可：

```
<meta http-equiv="refresh" content="5" />
```

2.2.4 网页的主体标记 body

网页所要显示的内容都放在网页的主体标记内，它是 HTML 文件的重点所在。后面章节所介绍的 HTML 标记都将放在这个标记内。然而，它并不仅仅是一个形式上的标记，它本身也可以控制网页的背景颜色或背景图像，这将在后面进行介绍。主体标记是以<body>开始、以</body>标记结束的，语法格式如下：

```
<body>
...
</body>
```



在构建 HTML 结构时，标记不允许交错出现，否则会造成错误。

在下列代码中，<body>开始标记出现在<head>标记内，这是错误的：

```
<html>
<head>
<title>标记测试</title>
<body>
</head>
</body>
</html>
```

2.2.5 页面注释标记<!-- -->

注释是在 HTML 代码中插入的描述性文本，用来解释该代码或提示其他信息。注释只出现在代码中，浏览器对注释代码不进行解释，并且在浏览器的页面中不显示。在 HTML 源代码中适当地插入注释语句是一种非常好的习惯，这对于设计者日后的代码修改、维护工作很有好处。另外，如果将代码交给其他设计者，其他人也能很快读懂前者所撰写的内容。

语法格式如下：

```
<!--注释的内容-->
```

注释语句元素由前后两半部分组成，前半部分由一个左尖括号、一个半角感叹号和两个连字符组成，后半部分由两个连字符和一个右尖括号组成：

```
<html>
<head>
   <title>标记测试</title>
</head>
<body>
   <!--这里是标题-->
   <h1>HTML 5 网页设计</h1>
</body>
</html>
```

页面注释不但可以对 HTML 中一行或多行代码进行解释说明，而且可以注释掉这些代码。如果希望某些 HTML 代码在浏览器中不显示，可以将这部分内容放在<!--和-->之间。例如，修改上述代码：

```
<html>
<head>
   <title>标记测试</title>
</head>
<body>
   <!--
   <h1>HTML 5 网页</h1>
   -->
</body>
</html>
```

修改后的代码将<h1>标记作为注释内容处理，在浏览器中将不会显示这部分内容。

2.3 HTML 5 语法的变化

为了兼容各个不统一的页面代码，HTML 5 的设计在语法方面做了以下变化。

2.3.1 标签不再区分大小写

标签不再区分大小写是 HTML 5 语法变化的重要体现。示例代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>大小写标签</title>
</head>
<body>
<P>这里的标签大小写不一样</p>
</body>
</html>
```

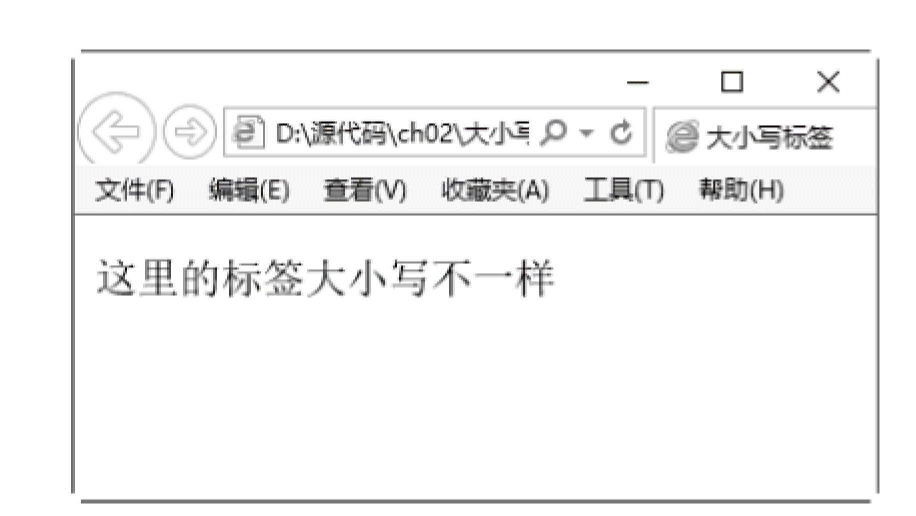


图 2-2 网页预览效果

在 IE 11.0 中预览，效果如图 2-2 所示。

虽然“<P>这里的标签大小写不一样</p>”中开始标记和结束标记不匹配，但是这完全符合 HTML 5 的规范。用户可以通过 W3C 提供的在线验证页面来测试上面的网页，验证网址为 http://validator.w3.org/。

2.3.2 允许属性值不使用引号

在 HTML 5 中，属性值不放在引号中也是正确的。例如以下代码片段：

```
<input checked="a" type="checkbox"/>
<input readonly type="text"/>
<input disabled="a" type="text"/>
```

上述代码片段与下面的代码片段效果是一样的：

```
<input checked=a type=checkbox/>
<input readonly type=text/>
<input disabled=a type=text/>
```



尽管 HTML 5 允许属性值可以不使用引号，但是仍然建议读者加上引号。因为如果某个属性的属性值中包含空格等容易引起混淆的属性值，此时可能会引起浏览器的误解。例如以下代码：

```
<img src=mm ch02/01.jpg />
```

此时浏览器就会误以为 src 属性的值就是 mm，这样就无法解析路径中的 01.jpg 图片。如果想正确解析到图片的位置，必须添加上引号。

2.3.3 允许部分属性值的属性省略

在 HTML 5 中，部分标志性属性的属性值可以省略。例如，以下代码是完全符合 HTML 5

规则的：

```
<input checked type="checkbox"/>
<input readonly type="text"/>
```

其中 checked="checked"省略为 checked，而 readonly="readonly"省略为 readonly。

在 HTML 5 中，可以省略属性值的属性如表 2-2 所示。

表 2-2　可以省略属性值的属性

属　性	省略属性值
checked	省略属性值后，等价于 checked="checked"
readonly	省略属性值后，等价于 readonly="readonly"
defer	省略属性值后，等价于 defer="defer"
ismap	省略属性值后，等价于 ismap="ismap"
nohref	省略属性值后，等价于 nohref="nohref"
noshade	省略属性值后，等价于 noshade="noshade"
nowrap	省略属性值后，等价于 nowrap="nowrap"
selected	省略属性值后，等价于 selected="selected"
disabled	省略属性值后，等价于 disabled="disabled"
multiple	省略属性值后，等价于 multiple="multiple"
noresize	省略属性值后，等价于 noresize="noresize"

2.4　必知必会——HTML 5 代码规范

很多学习网页设计的人员，对于 HTML 的代码规范知之甚少。作为一名优秀的网页设计人员，很有必要学习比较好的代码规范。HTML 5 代码规范介绍如下。

1. 使用小写标记名

在 HTML 5 中，元素名称可以大写也可以小写，推荐使用小写元素名。主要原因如下。

(1) 混合使用大小写元素名的代码是非常不规范的。

(2) 小写字母容易编写。

(3) 小写字母让代码看起来整齐而清爽。

(4) 网页开发人员往往使用小写，这样便于统一规范。

2. 标记要成对出现

在 HTML 5 中，一般标记都是成对出现的。除非是空的 HTML 标记。例如

```
<meta charset="utf-8">
```

也可以这么写：

```
<meta charset="utf-8" />
```

2.5　综合案例——符合 W3C 标准的 HTML 5 网页

通过本章的学习，读者了解到 HTML 5 较以前版本有了很大改变。本章就标记语法部分进行了详细的阐述。

下面将制作一个符合 W3C 标准的 HTML 5 网页，具体操作步骤如下。

step 01 启动 Dreamweaver CC，新建 HTML 文档，并单击文档工具栏中的“代码”按钮，切换至代码状态，如图 2-3 所示。

step 02 图 2-3 中的代码是 XHTML 1.0 格式的，尽管与 HTML 5 完全兼容，但是为了简化代码，可以将其修改成 HTML 5 规范的。修改文档说明部分、<html>标记部分和<meta>元信息部分。修改后，HTML 5 结构的代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>HTML 5 网页设计</title>
</head>

<body>
</body>
</html>
```

step 03 在网页主体中添加内容。例如，在 body 部分增加如下代码：

```
<!--白居易诗-->
<h1>续座右铭</h1>
<P>
千里始足下,<br>
高山起微尘。<br>
吾道亦如此,<br>
行之贵日新。<br>
</P>
```

step 04 保存网页，在 IE 11.0 中预览，效果如图 2-4 所示。



图 2-3　使用 Dreamweaver CC 新建 HTML 文档



图 2-4　网页的预览效果

2.6　跟我学上机——简单的 HTML 5 网页

下面制作一个简单的 HTML 5 网页，具体操作步骤如下。



step 01 打开记事本文件，在其中输入下述代码：

```
<!DOCTYPE html>
<html>
<head>
<title>简单的 HTML 5 网页</title>
</head>
<body>
  <h1>清明</h1>
  <P>
  清明时节雨纷纷,<br>
  路上行人欲断魂。<br>
  借问酒家何处有,<br>
  牧童遥指杏花村。<br>
  </P>
<img src="qingming.jpg">
</body>
</html>
```

step 02 保存网页，在 IE 11.0 中预览，效果如图 2-5 所示。



图 2-5 网页的预览效果

2.7 高 手 解 惑

疑问 1：在网页中，语言的编码方式有哪些？

答：在 HTML 5 网页中，<meta>标记的 charset 属性用于设置网页的内码语系，也就是字符集的类型。对于国内，经常要显示汉字，通常设置为 GB2312(简体中文)和 UTF-8 两种。英文是 ISO-8859-1 字符集。此外还有其他字符集，这里不再介绍。

疑问 2：网页中的基本标签是否必须成对出现？

答：在 HTML 5 网页中，大部分标签都是成对出现的。不过也有部分标签可以单独出现，<p/>、
、<img/>、<hr/>等。

第 3 章 HTML 5 与 HTML 4 的区别

HTML 5 中新增了大量元素与属性。这些新增的元素和属性使 HTML 5 的功能变得更加强大，使网页设计效果有了更多的实现可能。

重点案例效果



3.1 新增的主体结构元素

在 HTML 5 中，新增了几种新的与结构相关的元素，分别是 section 元素、article 元素、aside 元素、nav 元素和 time 元素。

3.1.1 案例 1——section 元素的使用

<section>标签定义文档中的节，如章节、页眉、页脚或文档中的其他部分。它可以与 h1、h2、h3、h4、h5、h6 等元素结合起来使用，标示文档结构。

section 标签的代码结构如下：

```
<section>
<h1>…</h1>
<p>…</p>
</section>
```

【例 3.1】 section 元素的使用(案例文件：ch03\3.1.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<section>
   <h2>section 元素使用方法</h2>
   <p> section 元素用于对网站或应用程序中页面上的内容进行分块。</p>
</section>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-1 所示，实现了内容的分块显示。

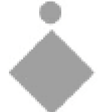


图 3-1 section 元素的使用

3.1.2 案例 2——article 元素的使用

<article>标签定义外部的内容。外部内容可以是来自一个外部的新闻提供者的一篇新的文章，或者来自 blog 的文本，或者是来自论坛的文本，或者是来自其他外部源内容。

article 标签的代码结构如下：



```
<article>
...
</article>
```

【例 3.2】article 元素的使用(案例文件：ch03\3.2.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<article>
   <header>
    <h1> HTML 5 教程</h1>
    <p>时间：<time pubdate="pubdate">2017-12-1</time></p>
   </header>
   <p>轻松学习 HTML 5 教程，就来</p>
<a href="http://www.yingda.com">www.yingda.com</a><br />
   <footer>
    <p><small>底部版权信息：yingda.com 公司所有</small></p>
   </footer>
  </article>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-2 所示，实现了外部内容的定义。



图 3-2 article 元素的使用

该实例讲述 article 元素的使用方法，在 header 元素中嵌入了文章的标题部分；在标题下部的 p 元素中，嵌入了一大段正文内容；在结尾处的 footer 元素中，嵌入了文章的著作权，作为脚注。整个示例的内容相对比较独立、完整，是因为这部分内容使用了 article 元素。

1. article 元素与 section 元素的区别

下面我们来介绍一下 article 元素与 section 元素的区别。

【例 3.3】article 元素与 section 元素的区别(案例文件：ch03\3.3.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<article>
     <h1>article 元素与 section 元素的使用方法</h1>
```

```
        <p>何时使用 article 元素？何时使用 section 元素…..</p>
        <section>
            <h2>article 元素使用方法</h2>
            <p>article 元素代表文档、页面或应用程序中独立的、完整的、可以独自被外部引用
的内容。</p>
        </section>
        <section>
            <h2>section 元素使用方法</h2>
            <p> section 元素用于对网站或应用程序中页面上的内容进行分块。</p>
        </section>
</article>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-3 所示，可以清楚地看到这两个元素的使用区别。



图 3-3　article 元素与 section 元素的区别

2．article 元素的嵌套

article 元素是可以嵌套使用的，内层的内容在原则上需要与外层的内容相关联。例如，一篇博客文章中，针对该文章的评论就可以使用嵌套 article 元素的方式，用来呈现评论的 article 元素被包含在表示整体内容的 article 元素里面。

【例 3.4】article 元素的嵌套(案例文件：ch03\3.4.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<article>
        <header>
            <h1>article 元素的嵌套</h1>
            <p>发表日期：<time pubdate="pubdate">2017/10/10</time></p>
        </header>
        <p>article 元素是什么？怎样使用 article 元素？…</p>
        <section>
            <h2>评论</h2>
            <article>
                <header>
                    <h3>发表者：唯一 </h3>
                    <p><time pubdate datetime="2011-12-23T:21-26:00">1 小时
```

```
                    前</time></p>
                </header>
                <p>这篇文章很不错啊，顶一下！</p>
            </article>
            <article>
                <header>
                    <h3>发表者：唯一</h3>
                    <p><time pubdate datetime="2013-2-20 T:21-26:00">1 小时
                       前</time></p>
                </header>
                <p>这篇文章很不错啊</p>
            </article>
        </section>
</article>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-4 所示。



图 3-4　article 元素的嵌套

这个实例中的代码比较完整，它添加了读者对文章的评论内容，实例内容分为几个部分。文章标题放在了 header 元素中，文章正文放在了 header 元素后面的 p 元素中，然后 section 元素把正文与评论进行了区分(是一个分块元素，用来把页面中的内容进行区分)。在 section 元素中嵌入了评论的内容，评论中每一个人的评论相对来说又是比较独立的、完整的。因此，对它们都使用一个 article 元素。在评论的 article 元素中，又可以分为标题与评论内容部分，分别放在 header 元素与 p 元素中。

3.1.3　案例 3——aside 元素的使用

aside 元素一般用来表示网站当前页面或文章的附属信息部分，它可以包含与当前页面或主要内容相关的广告、导航条、引用、侧边栏评论部分，以及其他区别于主要内容的部分。

aside 元素主要有以下两种使用方法。

(1) 被包含在 article 元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的相关资料、名称解释等。

aside 标签的代码结构如下：

```
<article>
  <h1>…</h1>
  <p>…</p>
  <aside>…</aside>
</article>
```

(2) 在 article 元素之外使用作为页面或站点全局的附属信息部分。最典型的是侧边栏，其中的内容可以是友情链接、博客中的其他文章列表、广告单元等。

aside 标签的代码结构如下：

```
<aside>
  <h2>…</h2>
  <ul>
    <li>…</li>
    <li>…</li>
  </ul>
  <h2>…</h2>
  <ul>
    <li>…</li>
    <li>…</li>
  </ul>
</aside>
```

【例 3.5】aside 元素的使用(案例文件：ch03\3.5.html)。

```
<!DOCTYPE html>
 <html>
 <head>
   <title>标题文件</title>
   <link rel="stylesheet" href="mystyles.css">
 </head>
 <body>
   <header>
     <h1>站点主标题</h1>
   </header>
   <nav>
     <ul>
       <li>主页</li>
       <li>图片</li>
       <li>音频</li>
     </ul>
   </nav>
   <section>
  </section>
   <aside>
     <blockquote>文章 1</blockquote>
     <blockquote>文章 2</blockquote>
   </aside>
</body>
 </html>
```

在 IE 11.0 中预览，效果如图 3-5 所示。



图 3-5　aside 元素的使用



<aside>元素可以位于示例页面的左边或右边，这个标签并没有预定义的位置。<aside>元素仅仅描述所包含的信息，而不反映结构。<aside>元素可位于布局的任意部分，用于表示任何非文档主要内容的部分。例如，可以在<section>元素中加入一个<aside>元素，甚至可以把该元素加入一些重要信息中，如文字引用。

3.1.4 案例4——nav元素的使用

<nav>用来将具有导航性质的链接划分在一起，使代码结构在语义化方面更加准确，同时对于屏幕阅读器等设备的支持也更好。

具体来说，nav元素可以用于以下这些场合。

(1) 传统导航条。现在主流网站上都有不同层级的导航条，其作用是将当前画面跳转到网站的其他主要页面上去。

(2) 侧边栏导航。现在主流博客网站及商品网站上都有侧边栏导航，其作用是将页面从当前文章或当前商品跳转到其他文章或其他商品页面上去。

(3) 页内导航。页内导航的作用是在本页面几个主要的组成部分之间进行跳转。

(4) 翻页操作。翻页操作是指在多个页面的前后页或博客网站的前后篇文章滚动。

(5) 除此之外，nav元素也可以用于其他所有用户觉得是重要的、基本的导航链接组中。

具体实现代码如下：

```
<nav>
<a href="…">Home</a>
<a href="…">Previous</a>
<a href="…">Next</a>
</nav>
```



如果文档中有“前后”按钮，则应该把它放到<nav>元素中。

一个页面中可以拥有多个<nav>元素，作为页面整体或不同部分的导航。下面给出一个代码实例。

【例3.6】 nav元素的使用(案例文件：ch03\3.6.html)。

```
<!DOCTYPE html>
<html>
<body>
<h1>技术资料</h1>
<nav>
   <ul>
      <li><a href="/">主页</a></li>
      <li><a href="/events">开发文档</a></li>
   </ul>
</nav>
<article>
  <header>
      <h1>HTML 5 与 CSS 3 的历史</h1>
      <nav>
         <ul>
```

```
            <li><a href="#HTML 5">HTML 5 的历史</a></li>
            <li><a href="#CSS 3">CSS 3 的历史</a></li>
          </ul>
      </nav>
   </header>
   <section id="HTML 5">
      <h1>HTML 5 的历史</h1>
      <p>讲述 HTML 5 的历史的正文</p>
      <footer>
      <p>
         <a href="?edit">以往版本</a> |
         <a href="?delete">当前现状</a> |
         <a href="?rename">未来前景</a>
      </p>
   </footer>
   </section>
   <section id="CSS 3">
      <h1>CSS 3 的历史</h1>
      <p>讲述 CSS 3 的历史的正文</p>
   </section>
   <footer>
      <p>
         <a href="?edit">以往版本</a> |
         <a href="?delete">当前现状</a> |
         <a href="?rename">未来前景</a>
      </p>
   </footer>
</article>
<footer>
   <p><small>版权所有：青花瓷</small></p>
</footer>
</body>
</html>
```

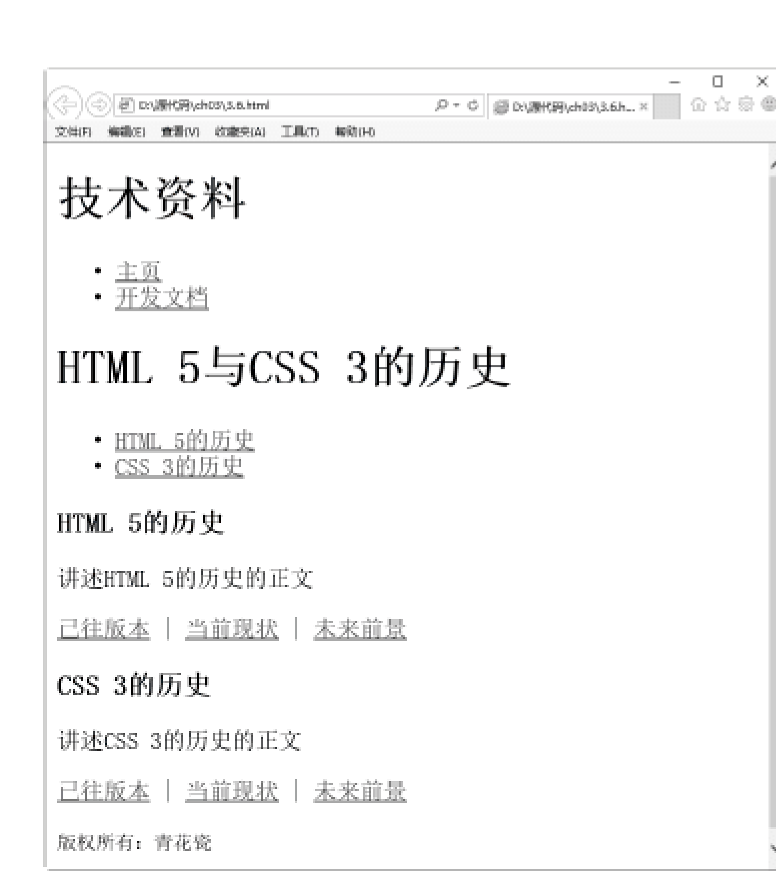
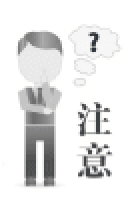


图 3-6　nav 元素的使用

在 IE 11.0 中预览，效果如图 3-6 所示。



在这个实例中，可以看到<nav>不仅可以用来作为页面全局导航，也可以放在<article>标签内，作为单篇文章内容的相关导航链接到当前页面的其他位置。

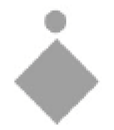


在 HTML 5 中不要用 menu 元素代替 nav 元素，menu 元素是用在一系列发出命令的菜单上的，是一种交互性元素，或者更确切地说是使用在 Web 应用程序中的。

3.1.5　案例 5——time 元素的使用

<time>是 HTML 5 新增加的一个标记，用于定义时间或日期。该元素可以代表 24 小时中的某一时刻，在表示时刻时，允许有时间差。在设置时间或日期时，只需要将该元素的属性 datetime 设为相应的时间或日期即可。

具体实现代码如下：



```
<p>
    <time>
    …
    </time>
</p>
<p>
    <time datetime=
    …
    </time>
</p>
```

【例 3.7】 time 元素的使用(案例文件：ch03\3.7.html)。

```
<!DOCTYPE html>
<html>
<body>
<h1>Time 元素</h1>
<p id="p1">
  <time datetime="2017-11-17">
今天是 2017 年 11 月 17 日
  </time>
 <p>
 <p id="p2">
  <time datetime="2017-11-17T17:00">
现在时间是 2017 年 11 月 17 日晚上 5 点
  </time>
 <p>
<p id="p3">
   <time datetime="2017-11-31">
    新款冬装将于今年年底上市
   </time>
 </p>
  <p id="p4">
    <time datetime="2017-11-15"
pubdate="true">
    本消息发布于 2017 年 11 月 15 日
    </time>
 </p>
</body>
</html>
```

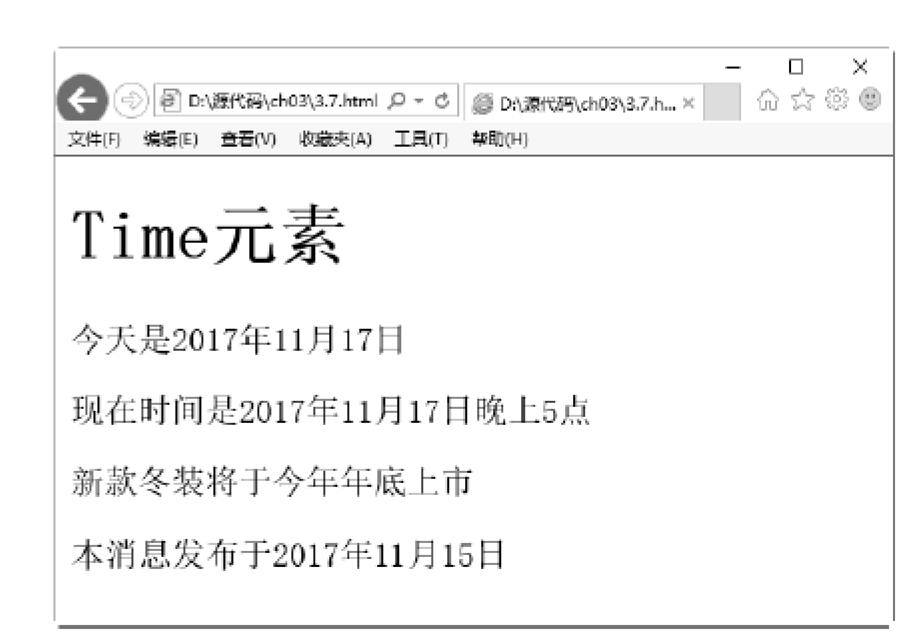
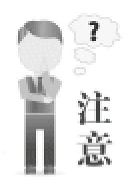


图 3-7　time 元素的使用

在 IE 11.0 中预览，效果如图 3-7 所示。

说明：

- <p>元素 id 号为 p1 中的<time>元素表示的是日期。页面在解析时，获取的是属性 datetime 中的值，而标记之间的内容只是用于显示在页面中。
- <p>元素 id 号为 p2 中的<time>元素表示的是日期和时间，它们之间使用字母 T 进行分隔。如果在整个日期与时间的后面加上一个字母 Z，则表示获取的是 UTC(世界统一时间)格式。
- <p>元素 id 号为 p3 中的<time>元素表示的是将来时间。
- <p>元素 id 号为 p4 中的<time>元素表示的是发布日期。



为了在文档中将这两个日期进行区分，在最后一个<time>元素中增加了pubdate属性，表示此日期为发布日期。



<time>元素中的可选属性 pubdate 表示时间是否为发布日期，它是一个布尔值，该属性不仅可以用于<time>元素，还可以用于<article>元素。

3.2 新增的非主体结构元素

在 HTML 5 中还新增了一些非主体的结构元素，如 header、hgroup、footer 等。

3.2.1 案例 6——header 元素的使用

header 元素是一种具有引导和导航作用的结构元素，通常用来放置整个页面或页面内的一个内容区块的标题，但也可以包含其他内容，如数据表格、搜索表单或相关的 logo 图片。

header 标签的代码结构如下：

```
<header>
<h1>…</h1>
<p>…</p>
</header>
```

在整个页面中的标题一般放在页面的开头，一个网页中没有限制 header 元素的个数，可以拥有多个，可以为每个内容区块加一个 header 元素。

【例 3.8】header 元素的使用(案例文件：ch03\3.8.html)。

```
<!DOCTYPE html>
<html>
<body>
<header>
  <h1>网页标题</h1>
</header>
<article>
  <header>
    <h1>文章标题</h1>
  </header>
  <p>文章正文</p>
</article>
</body>
</html>
```



图 3-8 header 元素的使用

在 IE 11.0 中预览，效果如图 3-8 所示。



在 HTML 5 中，一个 header 元素通常包括至少一个 headering 元素(h1～h6)，也可以包括 hgroup 元素、nav 元素，还可以包括其他元素。

3.2.2 案例 7——hgroup 元素的使用

<hgroup>标签用于对网页或区段(section)的标题进行组合。hgroup 元素通常会将 h1～h6

元素进行分组，譬如一个内容区块的标题及其子标题算一组。

hgroup 标签的使用代码如下：

```
<hgroup>
  <h1>…</h1>
  <h2>…t</h2>
</hgroup>
```

通常，如果文章只有一个主标题，是不需要 hgroup 元素的。如下这个实例就不需要使用 hgroup 元素。

【例 3.9】不使用 hgroup 元素(案例文件：ch03\3.9.html)。

```
<!DOCTYPE html>
<html>
<body>
<article>
    <header>
        <h1>文章标题</h1>
        <p><time datetime="2017-10-20">2017 年 10 月 20 日</time></p>
    </header>
    <p>文章正文</p>
</article>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-9 所示。



图 3-9　只有一个主标题

但是，如果文章有主标题，主标题下有子标题，就需要使用 hgroup 元素了。如下这个实例就需要使用 hgroup 元素。

【例 3.10】hgroup 元素的使用(案例文件：ch03\3.10.html)。

```
<!DOCTYPE html>
<html>
<body>
<article>
    <header>
        <hgroup>
```

```
            <h1>文章主标题</h1>
            <h2>文章子标题</h2>
        </hgroup>
        <p><time datetime="2017-10-20">2017 年 10 月 20 日</time></p>
    </header>
    <p>文章正文</p>
</article>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-10 所示。



图 3-10　主标题下有子标题

3.2.3　案例 8——footer 元素的使用

footer 元素可以作为其上层父级内容区块或是一个根区块的脚注。footer 通常包括其相关区块的脚注信息，如作者、相关阅读链接及版权信息等。

使用 footer 标签设置文档页脚的代码如下：

```
<footer>…</footer>
```

在 HTML 5 出现之前，网页设计人员使用下面的方式编写页脚。

【例 3.11】ul 元素的使用(案例文件：ch03\3.11.html)。

```
<!DOCTYPE html>
<html>
<body>
<div id="footer">
    <ul>
        <li>版权信息</li>
        <li>站点地图</li>
        <li>联系方式</li>
    </ul>
<div>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-11 所示。

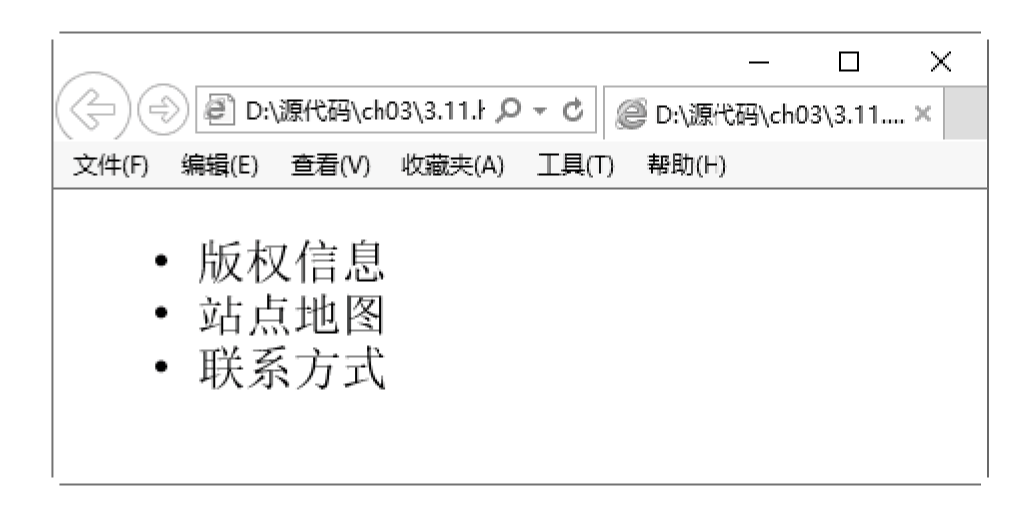


图 3-11　ul 元素的使用

但是到了 HTML 5 之后，这种方式将不再使用，而是使用更加语义化的 footer 元素来替代。

【例 3.12】 footer 元素的使用(案例文件：ch03\3.12.html)。

```
<!DOCTYPE html>
<html>
<body>
<footer>
    <ul>
        <li>版权信息</li>
        <li>站点地图</li>
        <li>联系方式</li>
    </ul>
</footer>
</body>
</html>
```

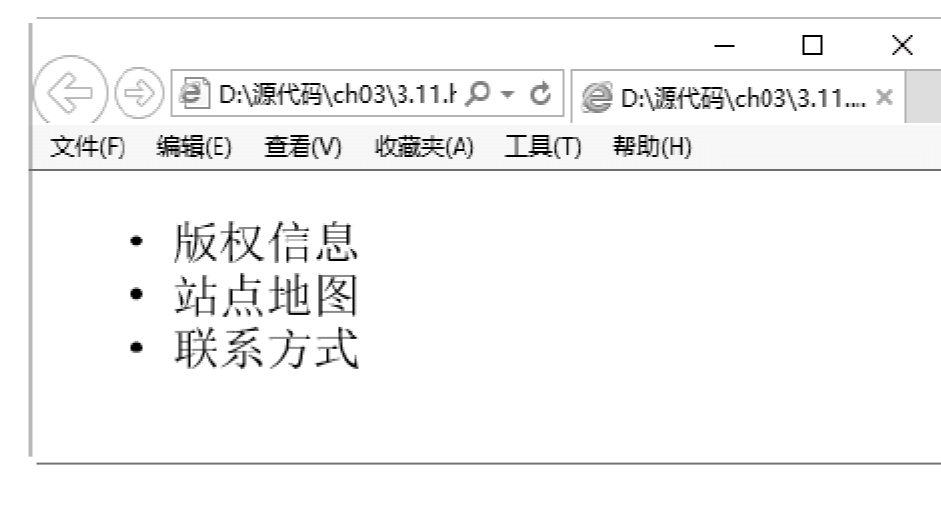


图 3-12　footer 元素的使用

在 IE 11.0 中预览，效果如图 3-12 所示。



与 header 元素一样，一个页面中并不限制 footer 元素的个数。同时，可以为 article 元素或 section 元素添加 footer 元素。

【例 3.13】 添加多个 footer 元素(案例文件：ch03\3.13.html)。

```
<!DOCTYPE html>
<html>
<body>
<article>
    文章内容
    <footer>
        文章的脚注
    </footer>
</article>
<section>
    分段内容
    <footer>
        分段内容的脚注
    </footer>
 </section>
</body>
</html>
```

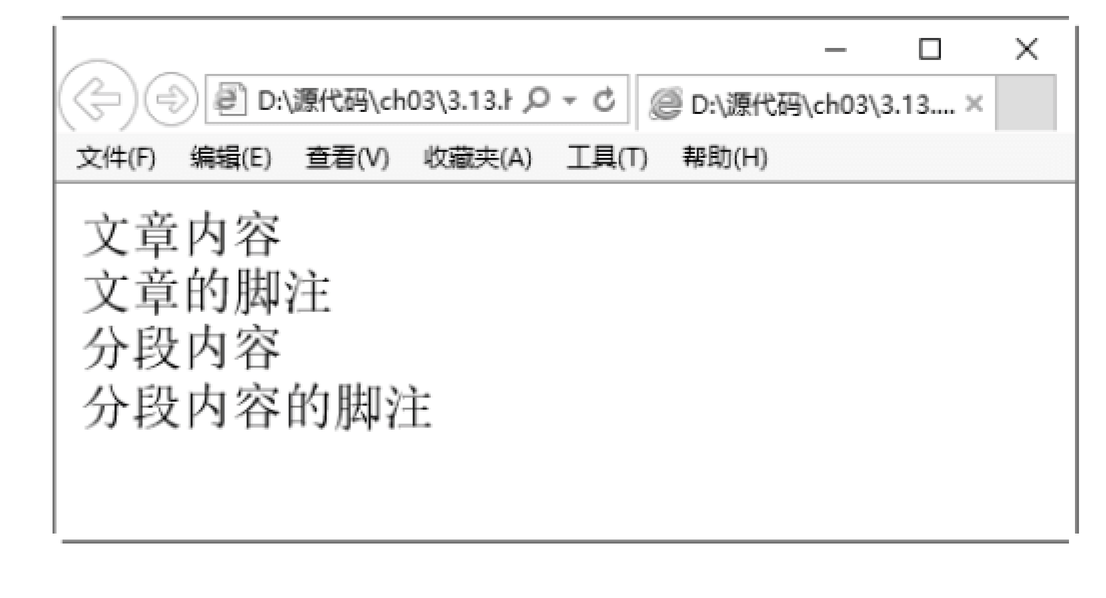


图 3-13　添加多个 footer 元素

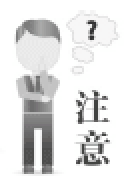
在 IE 11.0 中预览，效果如图 3-13 所示。

3.2.4　案例 9——figure 元素的使用

figure 元素是一种元素的组合，可以带有标题(可选)。figure 标签用来表示网页上一块独立的内容，将其从网页上移除后不会对网页上的其他内容产生影响。figure 所表示的内容可以是图片、统计图或代码示例。

figure 标签的实现代码如下：

```
<figure>
  <h1>…</h1>
  <p>…</p>
</figure>
```



使用 figure 元素时，需要 figcaption 元素为 figure 元素组添加标题。不过，一个 figure 元素内最多只允许放置一个 figcaption 元素，其他元素可无限放置。

1. 不带有标题的 figure 元素的使用

【例 3.14】不带有标题的 figure 元素的使用(案例文件：ch03\3.14.html)。

```
<!DOCTYPE HTML>
<html>
<head>
<title>不带有标题的 figure 元素</title>
</head>
<body>
   <figure>
      <img alt="images/logo.jpg"/>
   </figure>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-14 所示。



图 3-14 不带有标题的 figure 元素的使用

2. 带有标题的 figure 元素的使用

【例 3.15】带有标题的 figure 元素的使用(案例文件：ch03\3.15.html)。

```
<!DOCTYPE HTML>
<html>
<head>
<title>带有标题的 figure 元素</title>
</head>
<body>
   <figure>
      <img alt="images/logo.jpg"/>
   </figure>
   <figcaption>标题提示</figcaption>
</body>
</html>
```

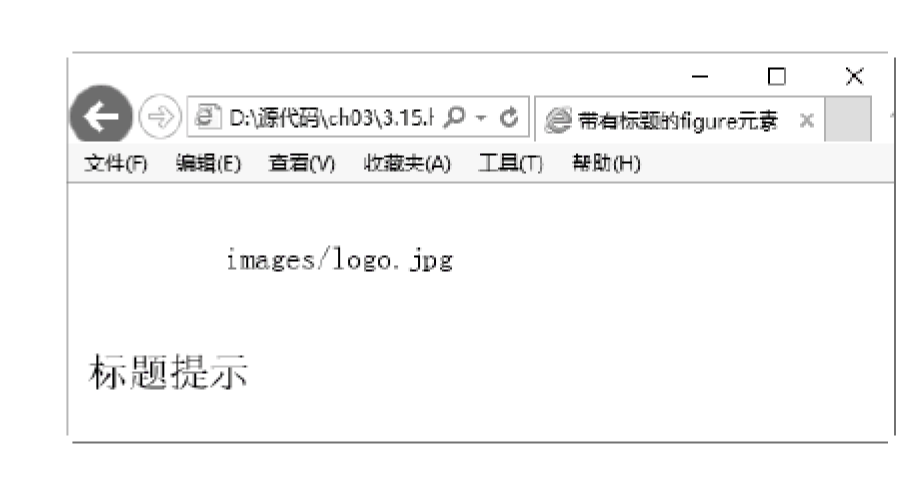
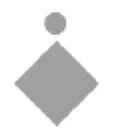


图 3-15 带有标题的 figure 元素的使用

在 IE 11.0 中预览，效果如图 3-15 所示。

3. 多张图片、同一标题的 figure 元素的使用

【例 3.16】多张图片、同一标题的 figure 元素的使用(案例文件：ch03\3.16.html)。



```
<!DOCTYPE HTML>
<html>
<head>
<title>多张图片，同一标题的 figure 元素
</title>
</head>
<body>
   <figure>
      <img alt="images/logo.jpg"/>
      <img alt="images/logo1.jpg"/>
      <img alt="images/logo2.jpg"/>
   </figure>
   <figcaption>标题提示</figcaption>
</body>
</html>
```

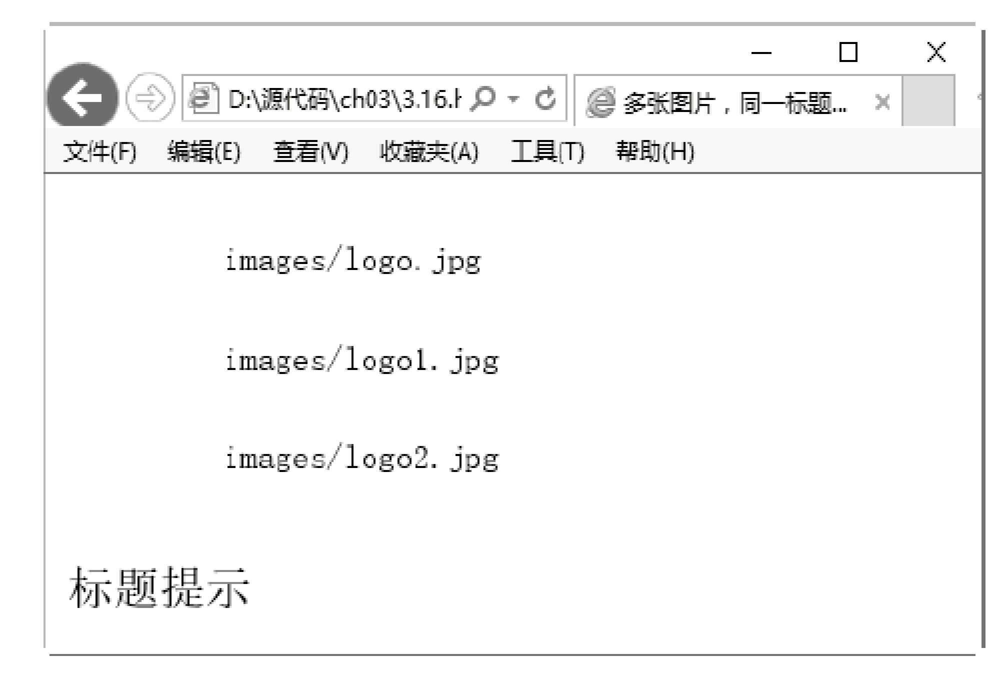


图 3-16 多张图片、同一标题的 figure 元素的使用

在 IE 11.0 中预览，效果如图 3-16 所示。

3.2.5 案例 10——address 元素的使用

address 元素用来在文档中呈现联系信息，包括文档作者或文档维护者的名字，以及其网站链接、电子邮箱、真实地址、电话号码等。

address 标签的实现代码如下：

```
<address>
      <a href=…>…</a>
      …
</address>
```

【例 3.17】address 元素的使用(案例文件：ch03\3.17.html)。

```
<!DOCTYPE html>
<html>
<body>
<address>
       <a href=http://blog.sina.com.cn/
        zhangsan>张三</a>
      <a href=http://blog.sina.com.cn/
        lisi>李四</a>
      <a href=http://blog.sina.com.cn/
        wanger>王二</a>
</address>
</body>
</html>
```

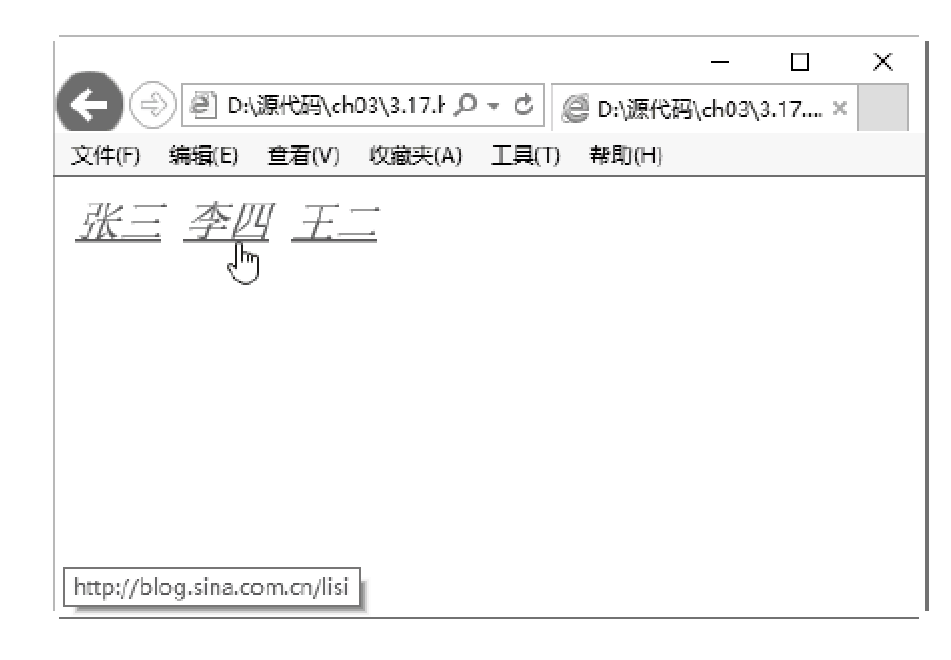


图 3-17 address 元素的使用

在 IE 11.0 中预览，效果如图 3-17 所示。

另外，address 元素不仅可以单独使用，还可以与 footer 元素、time 元素结合使用。

【例 3.18】address 元素与其他元素结合使用(案例文件：ch03\3.18.html)。

```
<!DOCTYPE html>
<html>
<body>
<footer>
```

```
    <div>
        <address>
          <a title="文章作者：张三" href="http://blog.sina.com.cn/zhangsan">
          张三</a>
        </address>
         发表于<time datetime="2017-11-17">2017 年 11 月 17 日</time>
    </div>
</footer>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-18 所示。



图 3-18 address 元素与其他元素结合使用

3.3 新增其他常用元素

除了结构元素外，在 HTML 5 中，还新增了其他元素，如 mark 元素、rp 元素、rt 元素、ruby 元素、progress 元素、command 元素、embed 元素、details 元素、summary 元素、datalist 元素等。

3.3.1 案例 11——mark 元素的使用

mark 元素主要用来在视觉上向用户呈现那些需要突出显示或高亮显示的文字。mark 元素的一个比较典型的应用就是在搜索结果中向用户高亮显示搜索关键词。其使用方法与<em>和<strong>有相似之处，但相比而言，HTML 5 中新增的 mark 元素在突出显示时更加随意与灵活。

HTML 5 中代码示例如下：

```
<p>… <mark>…</mark> …</p>
```

【例 3.19】mark 元素的使用(案例文件：ch03\3.19.html)。

在页面中，首先使用<h2>元素创建一个标题“优秀开发人员的素质”，然后通过<p>元素对标题进行阐述。在阐述的文字中，为了引起用户的注意，使用<mark>元素高亮处理字符“素质”“过硬”和“务实”。

具体的代码如下：

```
<!DOCTYPE html>
```

```
 <html>
 <head>
<title>mark 元素的使用</title>
</head>
 <body>
  <h2>优秀开发人员的<mark>素质</mark></h2>
  <p>一个优秀的 Web 页面开发人员，必须具有<mark>过硬</mark>的技术与<mark>务实</mark>
的专业精神 </p>
 </body>
 </html>
```

在 IE 11.0 中预览，效果如图 3-19 所示。

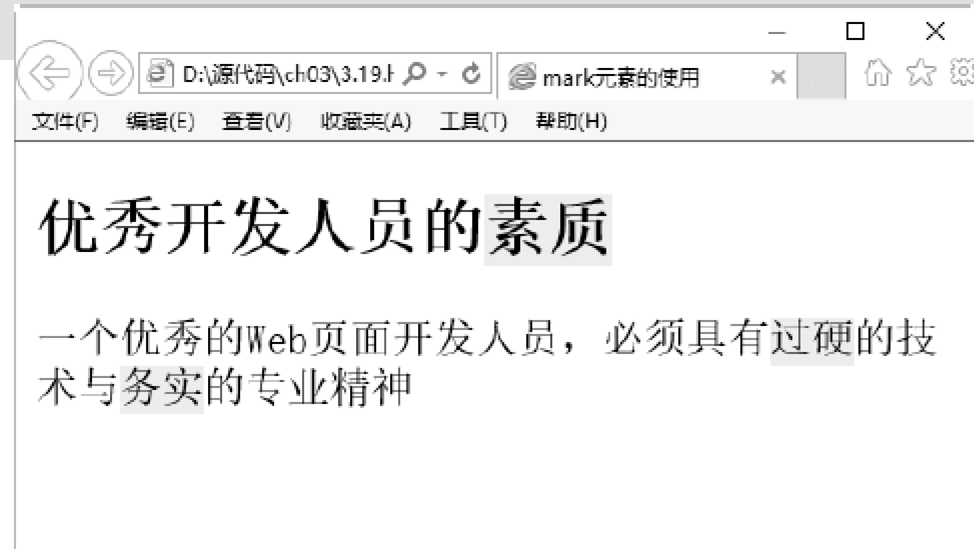


图 3-19 mark 元素的使用



mark 元素的这种高亮显示的特征，除用于文档中突出显示外，还常用于查看搜索结果页面中关键字的高亮显示，其目的主要是引起用户的注意。



虽然 mark 元素在使用效果上与 em 或 strong 元素有相似之处，但三者的出发点是不一样的。strong 元素是作者对文档中某段文字的重要性进行的强调；em 元素是作者为了突出文章的重点而进行的设置；mark 元素是数据展示时，以高亮形式显示某些字符，与原作者本意无关。

3.3.2 案例 12——rp 元素、rt 元素与 ruby 元素的使用

ruby 元素由一个或多个字符(需要一个解释/发音)和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持 ruby 元素时显示的内容。

rp 元素、rt 元素与 ruby 元素结合使用的代码如下。

```
<ruby>
   <rt><rp>(</rp>  <rp>)</rp></rt>
</ruby>
```

【例 3.20】使用 ruby 注释繁体字“漢”(案例文件：ch03\3.20.html)。

```
<!DOCTYPE html>
<html>
<body>
 <ruby>
    漢<rp>(</rp><rt>han</rt><rp>)</rp>
字<rp>(</rp><rt>zi</rt><rp>)</rp>
</ruby>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-20 所示。

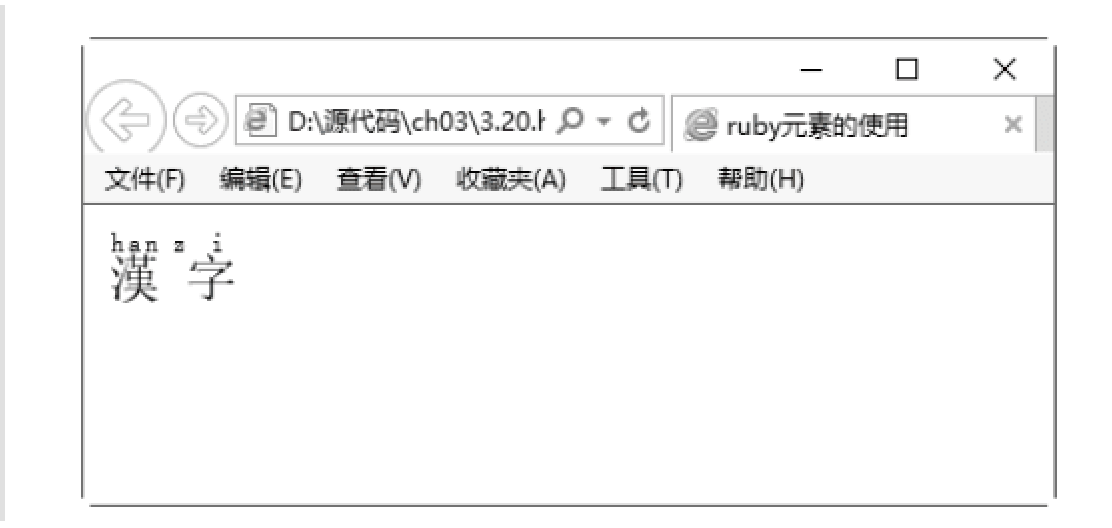


图 3-20 使用 ruby 注释繁体字“漢”



支持 ruby 元素的浏览器不会显示 rp 元素的内容。

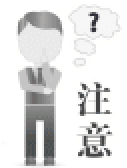
3.3.3 案例 13——progress 元素的使用

progress 元素表示运行中的进程。可以使用 progress 元素来显示 JavaScript 中耗费时间的函数进程。例如下载文件时，文件下载到本地的进度值可以通过该元素动态展示在页面中，展示的方式既可以使用整数(如 1～100)，也可以使用百分比(如 10%～100%)。

<progress>元素的属性及其描述如表 3-1 所示。

表 3-1　<progress>元素的属性及其描述

属　性	值	描　述
max	整数或浮点数	设置完成时的值，表示总体工作量
value	整数或浮点数	设置正在进行时的值，表示已完成的工作量



<progress>元素中设置的 value 值必须小于或等于 max 属性值，且两者都必须大于 0。

【例 3.21】使用 progress 元素表示下载进度(案例文件：ch03\3.21.html)。

```
<!DOCTYPE HTML>
<html>
<body>
    对象的下载进度:
    <progress  value="40"  max="100">>
    </progress>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-21 所示。

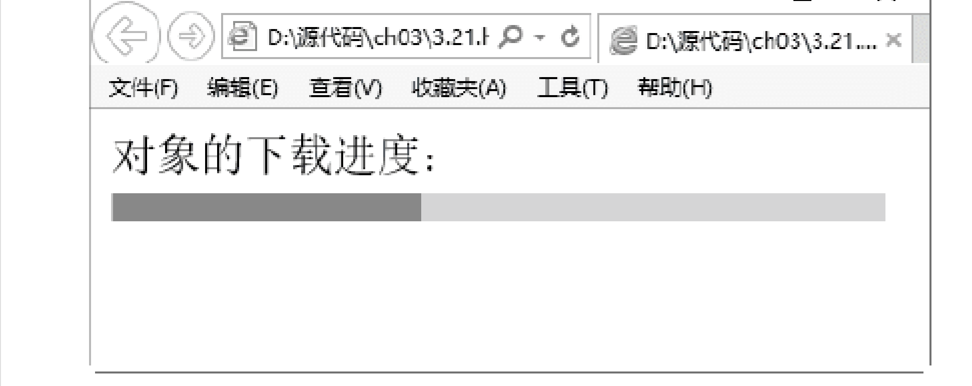


图 3-21　使用 progress 元素表示下载进度

3.3.4 案例 14——command 元素的使用

command 元素表示用户能够调用的命令，可以定义命令按钮，如单选按钮、复选框或按钮。

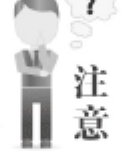
在 HTML 5 中使用 command 元素的代码如下：

```
<command type="command">…</command>
```

【例 3.22】使用 command 元素标记一个按钮(案例文件：ch03\3.22.html)。

```
<!DOCTYPE HTML>
<html>
<body>
   <menu>
     <command onclick="alert('Hello World')">Click Me!</command>
```

```
    </menu>
</body>
</html>
```



目前，主流浏览器都不支持<command>标签。只有 IE 9.0 支持<command>标签，其他之前版本或者之后版本的 IE 浏览器都不支持<command>标签。

在 IE 9.0 中预览，效果如图 3-22 所示。单击网页中的 Click Me 区域，将弹出提示信息框。

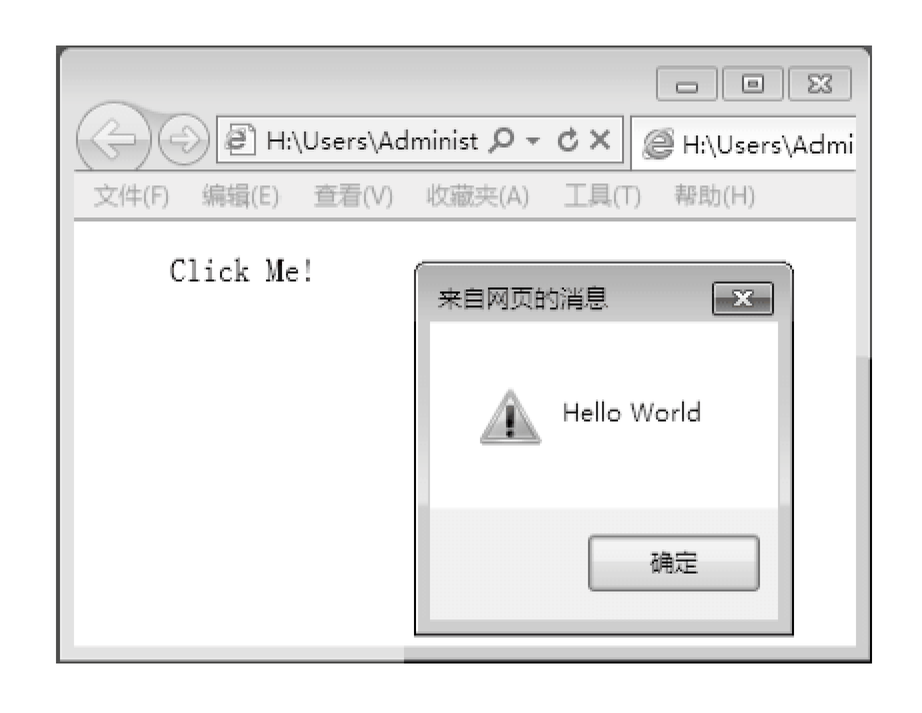


图 3-22　使用 command 元素标记一个按钮



只有当 command 元素位于 menu 元素内时，该元素才是可见的；否则，不会显示这个元素，但是可以用它规定键盘快捷键。

3.3.5　案例 15——embed 元素的使用

embed 元素用来插入各种多媒体，格式可以是 MIDI、WAV、AIFF、AU、MP3 等。

在 HTML 5 中使用 embed 元素的代码如下：

```
<embed src="…"/>
```

【例 3.23】使用 embed 元素插入动画(案例文件：ch03\3.23.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<embed src="images/飞翔的海鸟.swf"/>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-23 所示。



图 3-23　使用 embed 元素插入动画

3.3.6　案例 16——details 元素与 summary 元素的使用

details 元素表示用户要求得到并且可以得到的细节信息，与 summary 元素配合使用。summary 元素提供标题或图例。标题是可见的，用户单击标题时会显示出细节信息。summary 元素应该是 details 元素的第一个子元素。

在 HTML 5 中使用 details 元素和 summary 元素的代码如下。

```
<details>
    <summary>…</summary>
    …
</details>
```

【例 3.24】使用 details 元素制作简单页面(案例文件：ch03\3.24.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<details>
   <summary>苹果冰激凌</summary>
   <img src="images/冰激凌.jpg" alt="苹果冰激凌"/>
   <div>
      <h3> 材料：苹果 500g，白糖 150g，新鲜牛奶两瓶。</h3>
      <p>制作方法：将苹果洗净，去皮挖核，切成薄片，搅成浆状，放入白糖及 1000 克开水，加入
煮沸的牛奶，搅拌均匀，倒入盛器内冷却后置于冰箱冻结即成。
     </p>
   </div>
</details>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-24 所示。



图 3-24　使用 details 元素制作简单页面



在默认情况下，浏览器支持 details 元素，除了 summary 标签外的内容将会被隐藏。

3.3.7　案例 17——datalist 元素的使用

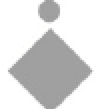
datalist 是用来辅助文本框的输入功能，它本身是隐藏的，与表单文本框中的 list 属性绑定，即将 list 属性值设置为 datalist 的 ID 号。它类似于 suggest 组件。目前只支持 Opera 浏览器。

在 HTML 5 中使用 datalist 元素的代码如下：

```
<datalist></datalist>
```

【例 3.25】使用 datalist 元素制作下拉列表框(案例文件：ch03\3.25.html)。

```
<!DOCTYPE HTML>
 <html>
```



```
<head>
    <title>datalist 测试</title>
</head>
<body>
<form action="#">
    <fieldset>
        <legend>请输入职业</legend>
        <input type="text" list="worklist">
        <datalist id="worklist">
            <option value="程序开发员"></option>
            <option value="系统架构师"></option>
            <option value="数据维护员"></option>
        </datalist>
    </fieldset>
</form>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 3-25 所示。

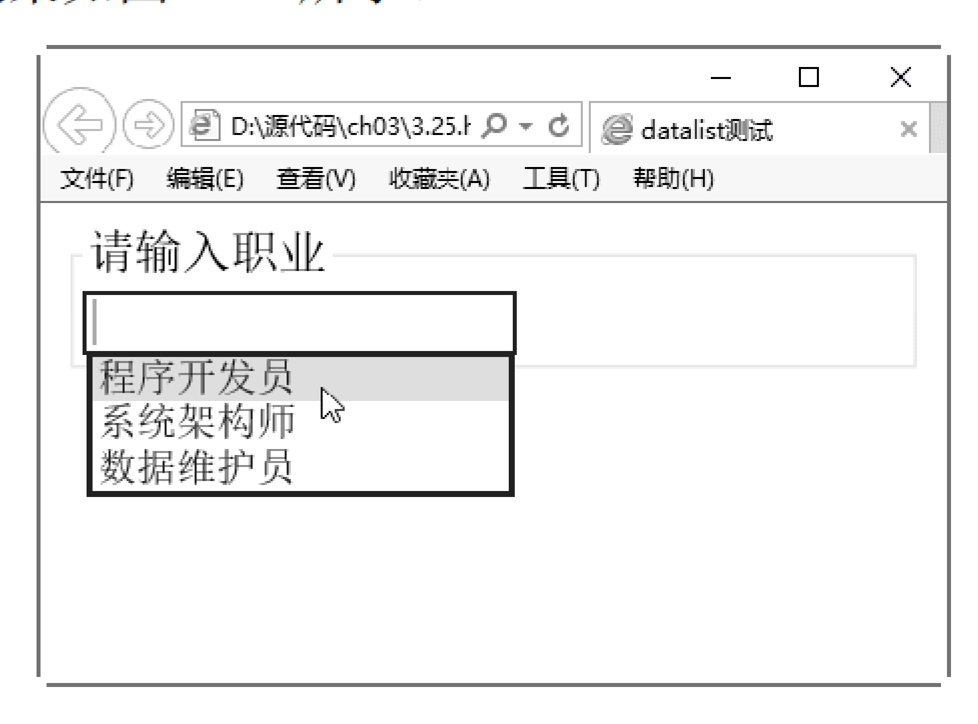


图 3-25 使用 datalist 元素制作下拉列表框

3.4 新增全局属性

在 HTML 5 中新增了许多全局属性。下面详细介绍常用的新增属性。

3.4.1 案例 18——contentEditable 属性的使用

contentEditable 属性是 HTML 5 中新增的标准属性，其主要功能是指定是否允许用户编辑内容。该属性有两个值：true 和 false。

contentEditable 属性为 true 表示可以编辑，false 表示不可编辑。如果没有指定值则会采用隐藏的 inherit(继承)状态，即如果元素的父元素是可编辑的，则该元素就是可编辑的。

【例 3.26】使用 contentEditable 属性的实例(案例文件：ch03\3.26.html)。

```
<!DOCTYPE html>
<HTML>
<head>
<title>contentEditable 属性示例</title>
```

```
</head>
<body>
<h3>对以下内容进行编辑内容</h3>
<ol contentEditable="true">
<li>列表一</li>
<li>列表二</li>
<li>列表三</li>
</ol>
</body>
</html>
```

使用 IE 11.0 预览，打开后可以在网页中输入相关内容，效果如图 3-26 所示。

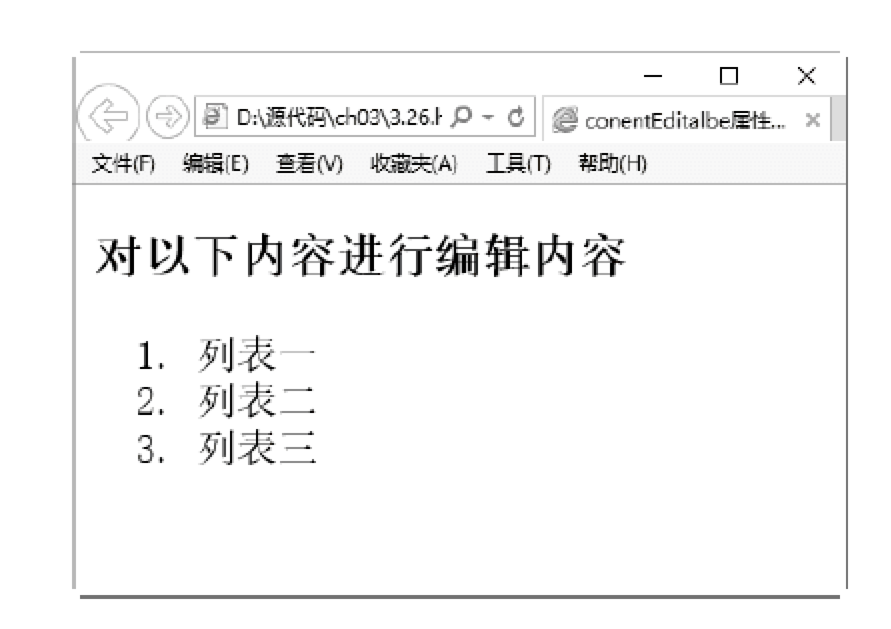
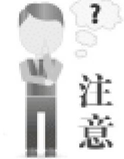


图 3-26　使用 contentEditable 属性的实例



注意　对内容进行编辑后，如果关闭网页，编辑的内容将不会被保存。如果想要保存其中的内容，只能把该元素的 innerHTML 发送到服务器端进行保存。

3.4.2　案例 19——spellcheck 属性的使用

spellcheck 属性是 HTML 5 中的新属性，规定是否对元素内容进行拼写检查。可对以下文本进行拼写检查：类型为 text 的 input 元素中的值(非密码)、textarea 元素中的值、可编辑元素中的值。

【例 3.27】使用 spellcheck 属性的实例(案例文件：ch03\3.27.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>hello, word</title>
</head>
<body>
<p contenteditable="true"
spellcheck="true">使用 spellcheck 属性，使段落内容可被编辑。</p>
</body>
</html>
```

使用 IE 11.0 预览，打开后可以在网页中输入相关内容，效果如图 3-27 所示。

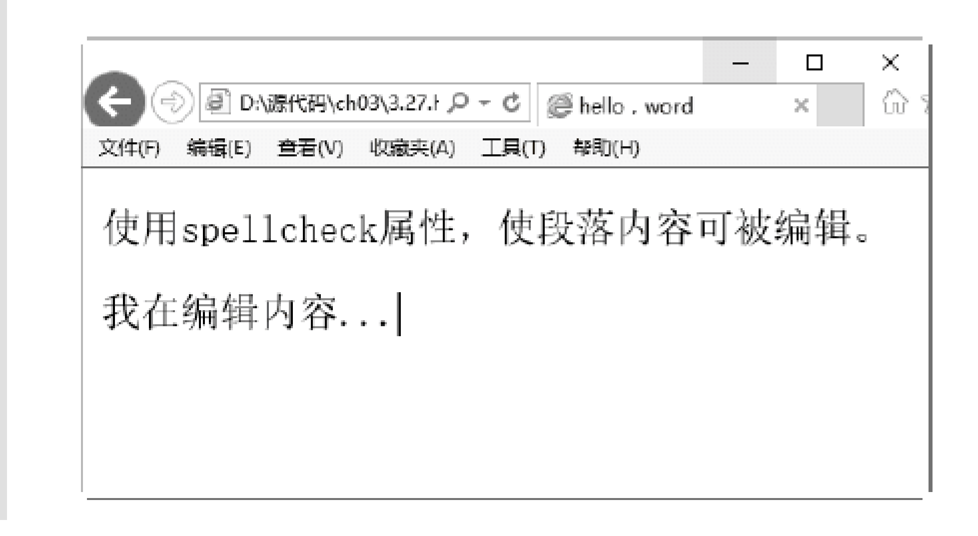


图 3-27　使用 spellcheck 属性的实例

3.4.3　案例 20——tabIndex 属性的使用

tabIndex 属性可设置或返回按钮的 Tab 键控制次序。打开页面，连续按 Tab 键，会在按钮之间进行切换，tabIndex 属性则可以记录显示切换的顺序。

【例 3.28】使用 tabIndex 属性的实例(案例文件：ch03\3.28.html)。

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function showTabIndex()
{
```

```
var bt1=document.getElementById('bt1').tabIndex;
var bt2=document.getElementById('bt2').tabIndex;
var bt3=document.getElementById('bt3').tabIndex;
document.write("Tab 切换按钮 1 的顺序： " + bt1);
document.write("<br />");
document.write("Tab 切换按钮 2 的顺序： " + bt2);
document.write("<br />");
document.write("Tab 切换按钮 3 的顺序： " + bt3);
}</script>
</head>
<body>
<button id="bt1" tabIndex="1">按钮 1</button><br />
<button id="bt2" tabIndex="2">按钮 2</button><br />
<button id="bt3" tabIndex="3">按钮 3</button><br />
<br />
<input type="button" onclick="showTabIndex()" value="显示切换顺序" />
</body>
</html>
```

使用 IE 11.0 预览，打开后多次按 Tab 键，使控制中心在几个按钮对象间切换，如图 3-28 所示。单击“显示切换顺序”按钮，显示出依次切换的顺序，如图 3-29 所示。



图 3-28　使用 tabIndex 属性的实例

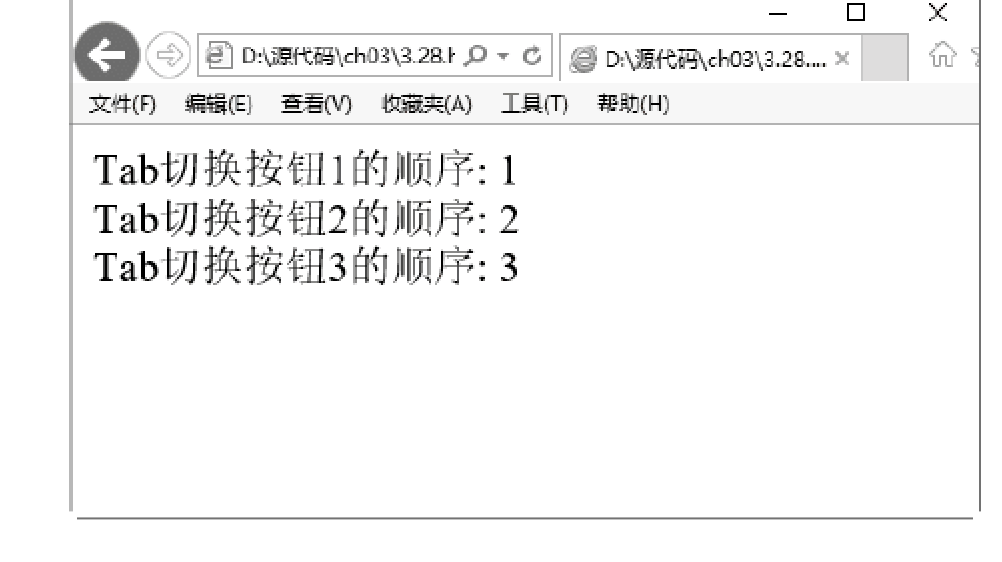


图 3-29　显示切换顺序

3.5　新增的其他属性

新增属性主要分为三大类：表单相关属性、链接相关属性和其他新增属性。具体内容介绍如下。

3.5.1　案例 21——表单相关属性的使用

新增的表单属性有很多，下面来分别进行介绍。

1．autocomplete

autocomplete 属性规定 form 或 input 域应该拥有自动完成功能。autocomplete 适用于<form>标签，以及以下类型的<input>标签：text、search、url、telephone、email、password、datepickers、range、color。

【例 3.29】使用 autocomplete 属性的实例(案例文件：ch03\3.29.html)。

```
<!DOCTYPE HTML>
<html>
```

```
<body>
<form action="demo form.asp" method="get" autocomplete="on">
   姓名:<input type="text" name="姓名" /><br />
   性别: <input type="text" sex="性别" /><br />
   邮箱: <input type="email" name="email" autocomplete="off" /><br />
   <input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-30 所示。

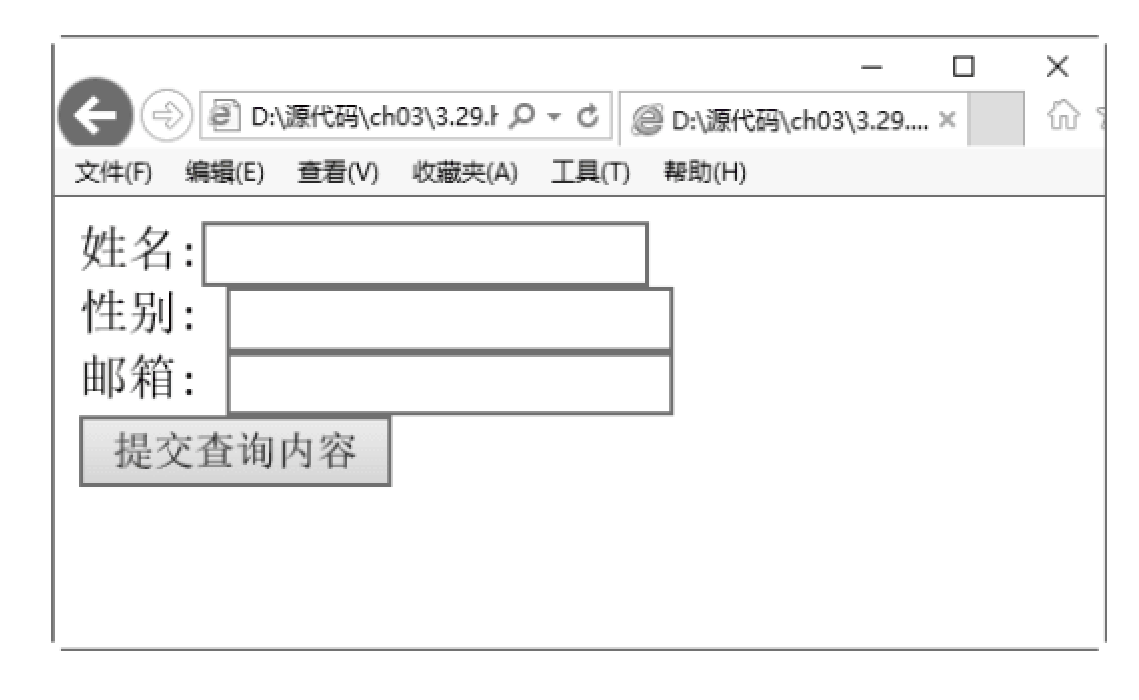


图 3-30　使用 autocomplete 属性的实例

2. autofocus

autofocus 属性规定在页面加载时，域自动地获得焦点。autofocus 属性适用于所有<input>标签的类型。

【例 3.30】使用 autofocus 属性的实例(案例文件：ch03\3.30.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo form.asp" method="get">
    用户名: <input type="text" name="user name" autofocus="autofocus" />
    <input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-31 所示。



图 3-31　使用 autofocus 属性的实例

3．form

form 属性规定输入域所属的一个或多个表单。form 属性适用于所有<input>标签的类型，必须引用所属表单的 id。

【例 3.31】使用 form 属性的实例(案例文件：ch03\3.31.html)

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo form.asp" method="get" id="user form">
    姓名:<input type="text" name="姓名" />
    <input type="submit" />
</form>
   性别: <input type="text" sex="性别" form="user form" />
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-32 所示。

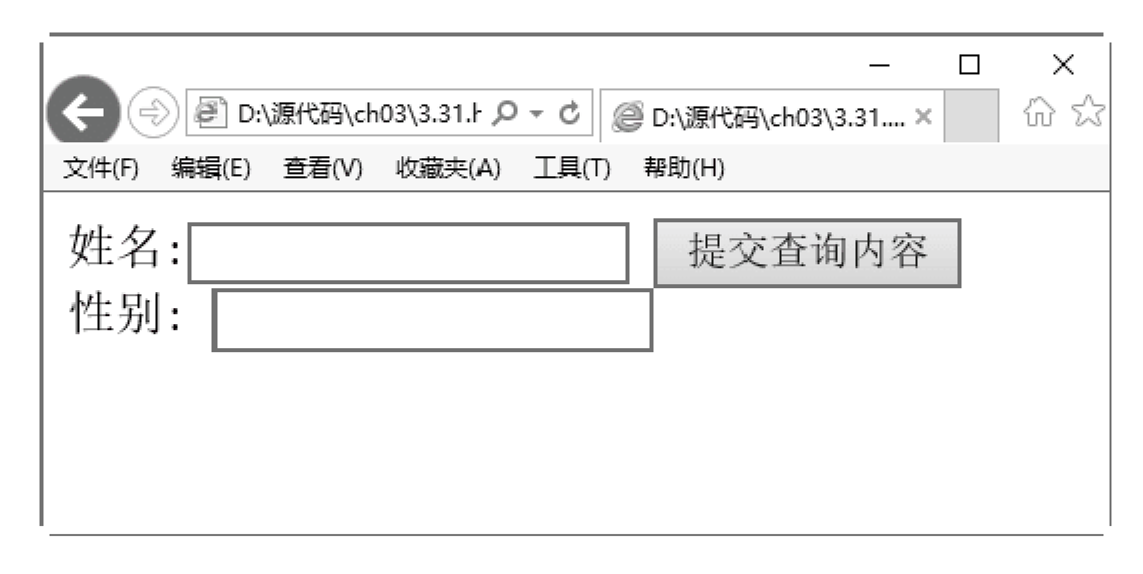


图 3-32　使用 form 属性的实例

4．form overrides

表单重写属性(form overrides attributes)允许重新设定 form 元素的某些属性。

表单重写属性如下。

(1) formaction：重写表单的 action 属性。

(2) formenctype：重写表单的 enctype 属性。

(3) formmethod：重写表单的 method 属性。

(4) formnovalidate：重写表单的 novalidate 属性。

(5) formtarget：重写表单的 target 属性。

表单重写属性适用于以下类型的<input>标签：submit 和 image。

【例 3.32】使用 form overrides 属性的实例(案例文件：ch03\3.32.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo_form.asp" method="get" id="user_form">
   邮箱: <input type="email" name="userid" /><br />
   <input type="submit" value="提交" /><br />
   <input type="submit" formaction="demo admin.asp" value="以管理员身份提交"
/><br />
   <input type="submit" formnovalidate="true" value="提交未经验证" /><br />
```

```
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-33 所示。

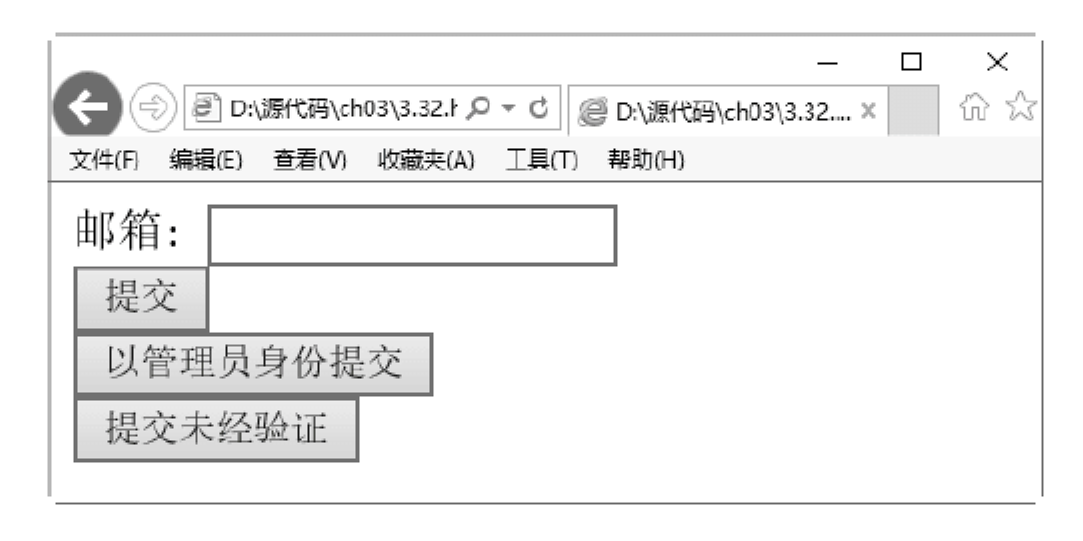


图 3-33　使用 form overrides 属性的实例

5．height 和 width

height 和 width 属性规定用于 image 类型的 input 标签的图像高度和宽度。height 和 width 属性只适用于 image 类型的<input>标签。

【例 3.33】使用 height 和 width 属性的实例(案例文件：ch03\3.33.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo_form.asp" method="get">
   用户名: <input type="text" name="user name" /><br />
   <input type="image" src="/images/按钮.jpg" width="99" height="99" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-34 所示。



图 3-34　使用 height 和 width 属性的实例

6．list

list 属性规定输入域的 datalist。datalist 是输入域的选项列表。list 属性适用于以下类型的<input>标签：text、search、url、telephone、email、date pickers、number、range 及 color。

【例 3.34】使用 list 属性的实例(案例文件：ch03\3.34.html)。



```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo_form.asp" method="get">
   主页: <input type="url" list="url_list" name="link" />
   <datalist id="url_list">
     <option label="baidu" value="http://www.baidu.com" />
     <option label="qq" value="http://www.qq.com" />
     <option label="Microsoft" value="http://www.microsoft.com" />
   </datalist>
<input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-35 所示。

图 3-35　使用 list 属性的实例

7．min、max 和 step

min、max 和 step 属性用于为包含数字或日期的 input 类型规定限定(约束)。max 属性规定输入域所允许的最大值；min 属性规定输入域所允许的最小值；step 属性为输入域规定合法的数字间隔(如果 step="3"，则合法的数是-3、0、3、6 等)。

min、max 和 step 属性适用于以下类型的<input>标签：date pickers、number 及 range。

【例 3.35】使用 min、max 和 step 属性的实例(案例文件：ch03\3.35.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo_form.asp" method="get">
    成绩: <input type="number" name="points" min="0" max="10" step="3"/>
<input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-36 所示。

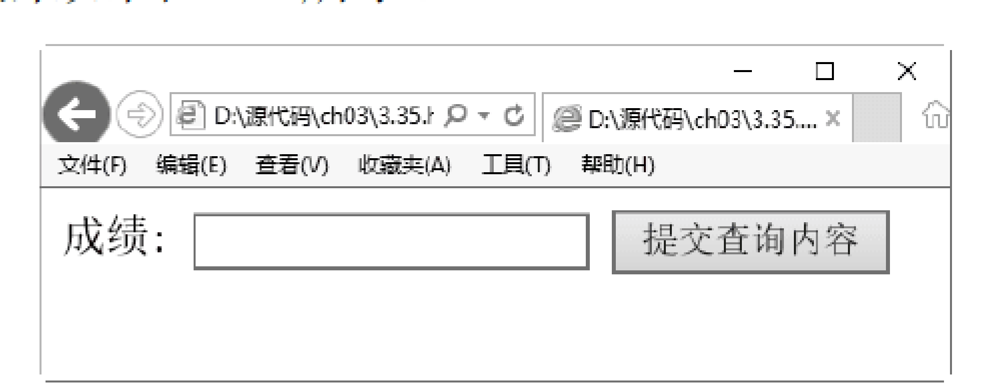


图 3-36　使用 min、max 和 step 属性的实例

8．multiple

multiple 属性规定输入域中可选择多个值。multiple 属性适用于以下类型的<input>标签：email 和 file。

【例 3.36】使用 multiple 属性的实例(案例文件：ch03\3.36.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo_form.asp" method="get">
   选择图片：<input type="file" name="img" multiple="multiple" />
<input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-37 所示。

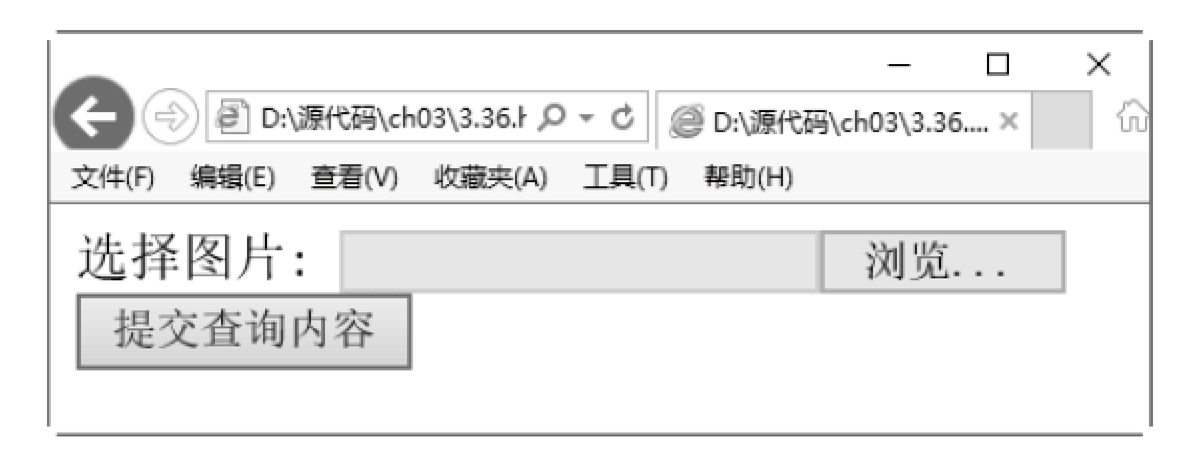


图 3-37 使用 multiple 属性的实例



单击“浏览”按钮，可以打开“选择要加载的文件”对话框，在其中选择要添加的图片信息。

9．pattern (regexp)

pattern 属性规定用于验证 input 域的模式(pattern)，适用于以下类型的<input>标签：text、search、url、telephone、email 及 password。

【例 3.37】使用 pattern 属性的实例(案例文件：ch03\3.37.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo_form.asp" method="get">
   电话区号：<input type="text" name="country_code" pattern="[A-z]{3}"
   title="Three letter country code" />
   <input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-38 所示。

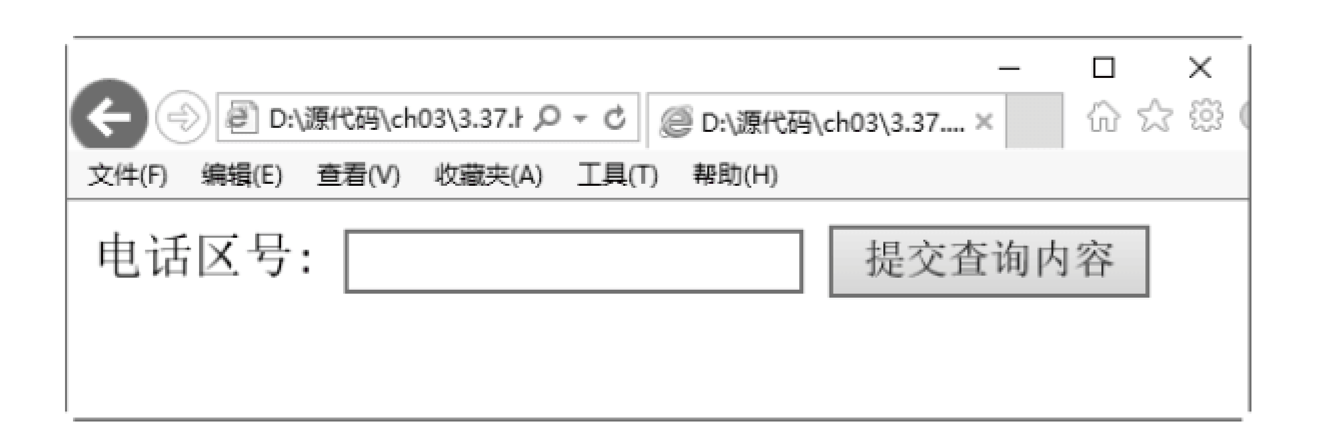


图 3-38 使用 pattern 属性的实例

10. placeholder

placeholder 属性提供一种提示(hint)，描述输入域所期待的值。placeholder 属性适用于以下类型的<input>标签：text、search、url、telephone、email 及 password。

【例 3.38】使用 placeholder 属性的实例(案例文件：ch03\3.38.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo form.asp" method="get">
   <input type="search" name="user search" placeholder="baidu" />
   <input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-39 所示。

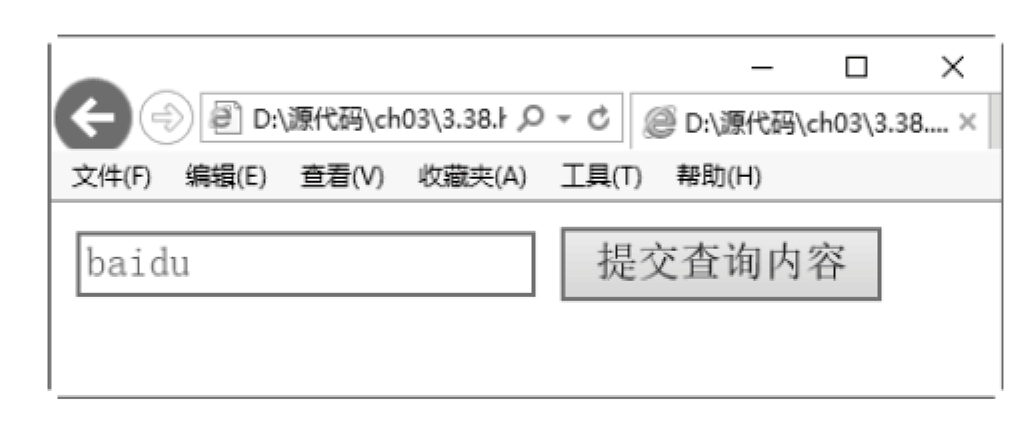


图 3-39 使用 placeholder 属性的实例

11. required

required 属性规定必须在提交之前填写输入域(不能为空)。required 属性适用于以下类型的<input>标签：text、search、url、telephone、email、password、date pickers、number、checkbox、radio 及 file。

【例 3.39】使用 required 属性的实例(案例文件：ch03\3.39.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form action="demo form.asp" method="get">
   姓名：<input type="text" name="usr name" required="required" />
   <input type="submit" />
</form>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-40 所示。

图 3-40　使用 required 属性的实例

3.5.2　案例 22——链接相关属性的使用

新增的与链接相关的属性介绍如下。

1. media

media 属性规定目标 URL 是为什么类型的媒介/设备进行优化的。该属性用于规定目标 URL 是为特殊设备(如 iPhone)、语音或打印媒介设计的。只能在 href 属性存在时使用。

【例 3.40】使用 media 属性的实例(案例文件：ch03\3.40.html)。

```
<!DOCTYPE HTML>
<html>
<body>
   <a href="www.baidu.com" media="print and (resolution:300dpi)">
     链接查询.
   </a>
</body>
</html>
```

使用 IE 9.0 预览，效果如图 3-41 所示。

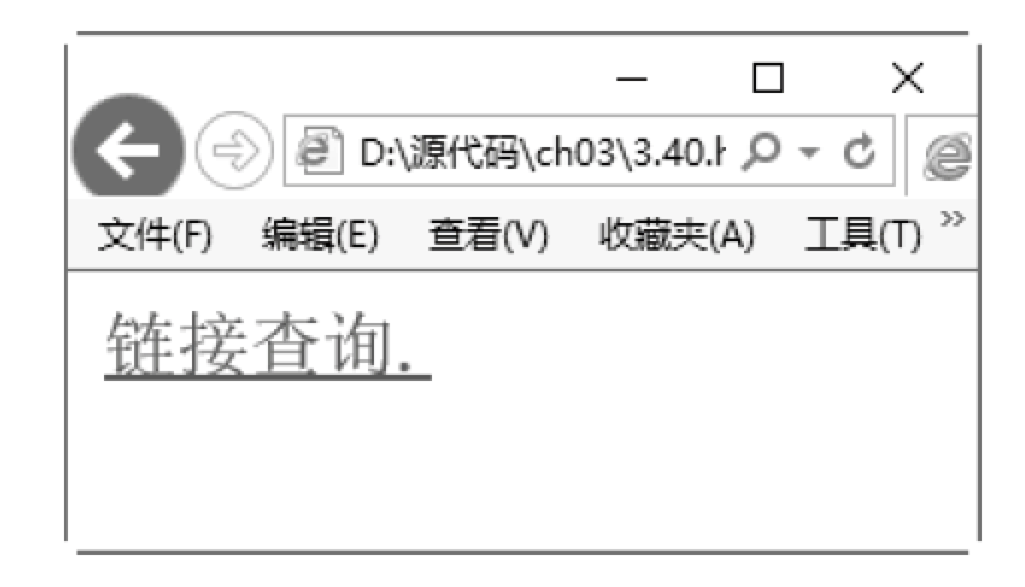


图 3-41　使用 media 属性的实例

2. type

在 HTML 5 中，为 area 元素增加了 type 属性，规定目标 URL 的 MIME 类型。仅在 href 属性存在时使用。

语法格式如下。

```
<input type="value">
```

3．sizes

为 link 元素增加了新属性 sizes。该属性可以与 icon 元素结合使用(通过 rel 属性)，该属性指定关联图标(icon 元素)的大小。

4．target

为 base 元素增加了 target 属性，主要目的是保持与 a 元素的一致性。

【例 3.41】使用 sizes 与 target 属性的实例(案例文件：ch03\3.41.html)。

```
<!DOCTYPE html>
<html>
<head>
    <link rel="icon" href="demo icon.ico" type="image/gif" sizes="16x16" />
</head>
<body>
    <h2>Hello world!</h2>
    <p>打开<a href="2.40.html" target=" blank">新链接</a>窗口。</p>
</body>
</html>
```

使用 IE 11.0 预览，效果如图 3-42 所示。



图 3-42　使用 sizes 与 target 属性的实例

3.5.3　案例 23——其他新增属性的使用

除了以上介绍的与表单和链接相关的属性外，HTML 5 还增加了其他属性，如表 3-2 所示。

表 3-2　HTML 5 增加的其他属性

属　性	隶 属 于	意　义
reversed	ol 元素	指定列表倒序显示
charset	meta 元素	为文档字符编码的指定提供了一种良好的方式
type	menu 元素	让菜单可以以上下文菜单、工具条与列表菜单 3 种形式出现
label	menu 元素	为菜单定义一个可见的标注
scoped	style 元素	用来规定样式的作用范围，譬如只对页面上某个数起作用
async	script 元素	定义脚本是否异步执行

续表

属　性	隶 属 于	意　义
manifest	html 元素	开发离线 Web 应用程序时它与 API 结合使用，定义一个 URL，在这个 URL 上描述文档的缓存信息
sandbox、srcdoc 与 seamless	iframe 元素	用来提高页面安全性，防止不信任的 Web 页面执行某些操作

3.6　HTML 5 废除的属性

在 HTML 5 中废除了很多不需要再使用的属性，这些属性将采用其他属性或方案进行替代，具体内容如表 3-3 所示。

表 3-3　HTML 5 中废除的属性

废除的属性	使用该属性的元素	在 HTML 5 中代替的方案
rev	Link，a	rel
charset	Link，a	在被链接的资源中使用 HTTP content-type 头元素
shape，coords	a	使用 area 元素代替 a 元素
longdesc	img，iframe	使用 a 元素链接到较长描述
target	link	多余属性，被省略
nohref	area	多余属性，被省略
profile	head	多余属性，被省略
version	html	多余属性，被省略
name	img	id
scheme	meta	只为某个表单域使用 scheme
Archive，classid，codebase，codetype，declare，standby	object	使用 data 与 type 属性类调用插件。需要使用这些属性来设置参数时，使用 param 属性
valuetype，type	param	使用 name 与 value 属性，不声明值的 MIME 类型
axis，abbr	td，th	使用以明确简洁的文字开头，后跟详述文字的形式。可以对更详细内容使用 title 属性，来使单元格的内容变得简短
scope	td	在被链接的资源中使用 HTTP Content-type 头元素
align	caption，input，legend，div，h1，h2，h3，h4，h5，h6，p	使用 CSS 样式表进行替代

续表

废除的属性	使用该属性的元素	在 HTML 5 中代替的方案
Alink，link，text，vlink，background，bgcolor	body	使用 CSS 样式表进行替代
Align，bgcolor，border，cellpadding，cellspacing，Frame，rules，width	table	使用 CSS 样式表进行替代
Align，char，charoff，height，nowrap，valign	tbody，thead，tfoot	使用 CSS 样式表进行替代
align，bgcolor，char，charoff，height，nowrap，valign，width	td，th	使用 CSS 样式表进行替代
Align，bgcolor，char，charoff，valign	tr	使用 CSS 样式表进行替代
Align，char，charoff，valign，width	Col，colgroup	使用 CSS 样式表进行替代
Align，border，hspace，vspace	object	使用 CSS 样式表进行替代
clear	br	使用 CSS 样式表进行替代
Compact，type	ol，ul，li	使用 CSS 样式表进行替代
compact	dl	使用 CSS 样式表进行替代
compact	menu	使用 CSS 样式表进行替代
width	pre	使用 CSS 样式表进行替代
Align，hspace，vspace	img	使用 CSS 样式表进行替代
Align，noshade，size，width	hr	使用 CSS 样式表进行替代
Align，frameborder，scrollingmarginheight，marginwidth	iframe	使用 CSS 样式表进行替代
autosubmit	menu	

3.7　高 手 解 惑

疑问 1：HTML 5 中的单标记和双标记书写方法有哪些？

答：HTML 5 中的标记分为单标记和双标记。单标记是指没有结束标记的标签；双标记既有开始标签又包含结束标签。

单标记是不允许写结束标记的元素，只允许使用“<元素 />”的形式进行书写。例如“
…</br>”的书写方式是错误的，正确的书写方式为
。当然，在 HTML 5 之前版本中
的这种书写方法可以被沿用。HTML 5 中不允许写结束标记的元素有 area、base、br、col、command、embed、hr、img、input、keygen、link、meta、param、source、track、wbr。

部分双标记可以省略结束标记。HTML 5 中允许省略结束标记的元素有 li、dt、dd、p、rt、rp、optgroup、option、colgroup、thead、tbody、tfoot、tr、td、th。

HTML 5 中有些元素还可以完全被省略。即使这些标记被省略了，该元素还是以隐式的方式存在的。HTML 5 中允许省略全部标记的元素有 html、head、body、colgroup、tbody。

疑问 2：新增属性 Target 在 HTML 4.01 与 HTML 5 之间的差异有哪些？

答：在 HTML 5 中，不再允许把框架名称设定为目标，因为不再支持 frame 和 frameset。self、parent 及 top 这 3 个值大多数时候与 iframe 一起使用。

第 II 篇

核 心 技 术

➘ 第 4 章　设计网页文本内容
➘ 第 5 章　设计网页列表与段落
➘ 第 6 章　HTML 5 网页中的图像
➘ 第 7 章　使用 HTML 5 建立超链接
➘ 第 8 章　使用 HTML 5 创建表单
➘ 第 9 章　使用 HTML 5 创建表格
➘ 第 10 章　HTML 5 中的音频和视频
➘ 第 11 章　使用 HTML 5 绘制图形



第 4 章 设计网页文本内容

网页文本是网页中最主要也是最常用的元素。网页文本的内容包括标题文字、普通文字、段落文字、水平线等。本章介绍如何使用 HTML 5 设计网页文本内容。



4.1　标题文字的建立

在 HTML 文档中，文本的结构除了以行和段出现之外，还可以作为标题存在。通常一篇文档最基本的结构就是由若干不同级别的标题和正文组成的。

4.1.1　案例 1——标题文字标记

HTML 文档中包含有各种级别的标题。各种级别的标题由<h1>到<h6>元素来定义。<h1>至<h6>标题标记中的字母 h 是英文 headline(标题行)的简称。其中<h1>代表 1 级标题，级别最高，文字也最大，其他标题元素依次递减，<h6>级别最低。

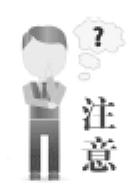
【例 4.1】标题标记的使用(案例文件：ch04\4.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>标题文字</title>
</head>
<body>
<h1>这里是 1 级标题</h1>
<h2>这里是 2 级标题</h2>
<h3>这里是 3 级标题</h3>
<h4>这里是 4 级标题</h4>
<h5>这里是 5 级标题</h5>
<h6>这里是 6 级标题</h6>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-1 所示。



图 4-1　标题标记的使用



作为标题，它们的重要性是有区别的，其中<h1>标题的重要性最高，<h6>的重要性最低。



4.1.2 案例 2——标题文字的对齐方式

标题文字的对齐方式主要有居左、居中、居右和两端对齐，其中两端对齐方式不经常使用。

【例 4.2】标题文字的对齐方式(案例文件：ch04\4.2.html)。

```
<!DOCTYPE html>
<html>
<body>
<h1 align="center">这里是 1 级标题 居中对齐</h1>
<h2 align="left">这里是 2 级标题 居左对齐</h2>
<h3 align="right">这里是 3 级标题 居右对齐</h3>
<p>上面的标题在页面中进行了各种对齐方式的排列。上面的标题在页面中进行了各种对齐方式的排列。上面的标题在页面中进行了各种对齐方式的排列。</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-2 所示。



图 4-2 标题文字的对齐方式

4.2 设置文字格式

一个杂乱无序、堆砌而成的网页，会使人感觉枯燥无味，而一个美观大方的网页，会让人有美轮美奂、流连忘返的感觉。下面介绍如何设置网页文字的格式。

4.2.1 案例 3——设置文字字体

font-family 属性用于指定文字字体类型，如宋体、黑体、隶书、Times New Roman 等，即在网页中，展示字体不同的形状。语法格式如下：

```
style="font-family:黑体"
style="font-family:华文彩云,黑体,宋体"
```

从语法格式上可以看出，font-family 有两种声明方式。第一种声明方式使用 name 字体名称，按优先顺序排列，以逗号隔开，如果字体名称包含空格，则应使用引号括起。第二种声明方式使用所列出的字体序列名称。如果使用 fantasy 序列，将提供默认字体序列。第一种声明方式比较常用。

【例 4.3】设置文字字体及对齐方式(案例文件：ch04\4.3.html)。

```
<!DOCTYPE html>
<html>
<head><title>字体</title>
</head>
<body>
<p style="font-family:黑体" align=center>北国风光，千里冰封。</p>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 4-3 所示，可以看到文字为黑体并居中显示。

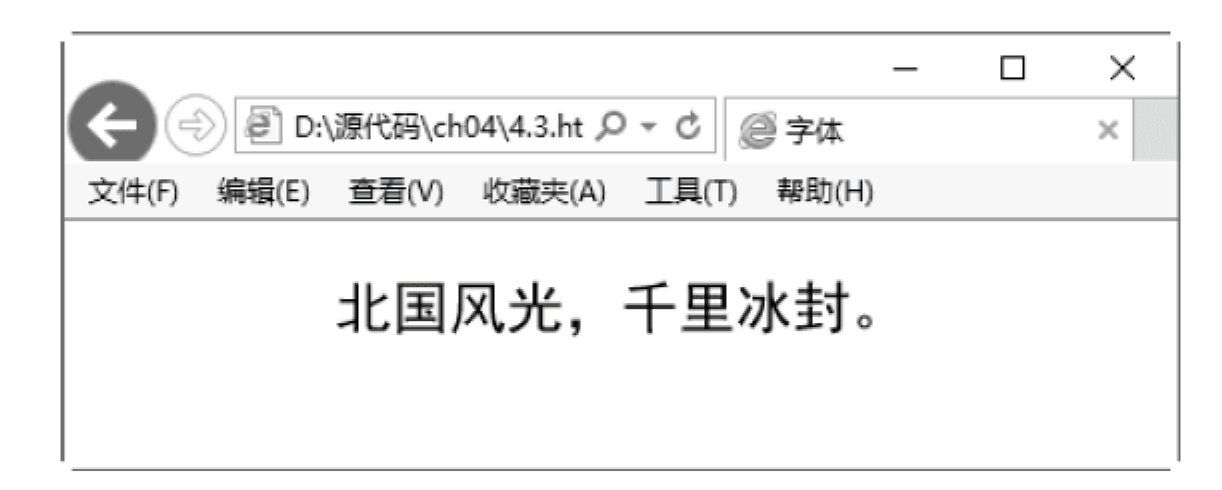


图 4-3　设置文字字体及对齐方式

在字体显示时，如果指定一种特殊字体类型，而在浏览器或者操作系统中该类型不能正确获取，可以通过 font-family 预设多种字体类型。font-family 属性可以预置多个供页面使用的字体类型，即字体类型序列，其中每种字体类型之间使用逗号隔开。如果前面的字体类型不能够正确显示，则系统将自动选择后一种字体类型，依次类推。所以，在设计页面时，一定要考虑字体的显示问题。为了保证页面达到预期效果，最好提供多种字体类型，而且最好以最基本的字体类型作为最后一个。

其样式设置如下。

```
font-family:华文彩云,黑体,宋体
```

当 font-family 属性值中的字体类型由多个字符串和空格组成，如 Times New Roman，那么，该值就需要使用双引号引起来。

```
font-family: "Times New Roman"
```

4.2.2　案例 4——设置字号

在一个网页中，标题通常使用较大字体显示，用于吸引人注意，小字体用来显示正常内容，大小字体结合形成网页，既吸引了人的眼球，又提高了阅读速度。

在 HTML 5 新规定中，通常使用 font-size 设置文字大小。其语法格式如下：

```
Style="font-size : 数值| inherit | xx-small | x-small | small | medium |
large | x-large | xx-large | larger | smaller | length"
```

其中，通过数值来定义字体大小，例如用 font-size:10px 的方式定义字体大小为 10 像素。此外，还可以通过 medium 之类的参数定义字体的大小，其参数含义如表 4-1 所示。

表 4-1　设置字体大小的参数

参　数	说　明
xx-small	绝对字体尺寸。根据对象字体进行调整。最小
x-small	绝对字体尺寸。根据对象字体进行调整。较小
small	绝对字体尺寸。根据对象字体进行调整。小
medium	默认值。绝对字体尺寸。根据对象字体进行调整。正常
large	绝对字体尺寸。根据对象字体进行调整。大
x-large	绝对字体尺寸。根据对象字体进行调整。较大
xx-large	绝对字体尺寸。根据对象字体进行调整。最大
larger	相对字体尺寸。相对于父对象中字体尺寸进行相对增大。使用成比例的 em 单位计算
smaller	相对字体尺寸。相对于父对象中字体尺寸进行相对减小。使用成比例的 em 单位计算
length	百分数或由浮点数字和单位标识符组成的长度值，不可为负值。其百分比取值是基于父对象中字体的尺寸

【例 4.4】设置文字字号(案例文件：ch04\4.4.html)。

```
<!DOCTYPE html>
<html>
<head><title>字号</title></head>
<body>
<p style="font-size:20pt">上级标记大小</p>
<p style="font-size:small">小</p>
<p style="font-size:larger">大</p>
<p style="font-size:x-small">小</p>
<p style="font-size:x-larger">大</p>
<p style="font-size:50%">子标记</p>
<p style="font-size:25pt">子标记</p>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 4-4 所示，可以看到网页中文字被设置成不同的大小，其设置方式采用了绝对数值、关键字、百分比等形式。

在上述例子中，font-size 字体大小为 50%时，其比较对象是上一级标签中的 20pt。同样，我们还可以使用 inherit 值，直接继承上级标记的字体大小。例如：

```
<p style="font-size:50pt">上级标记</p>
<p style="font-size: inherit ">继承</p>
```

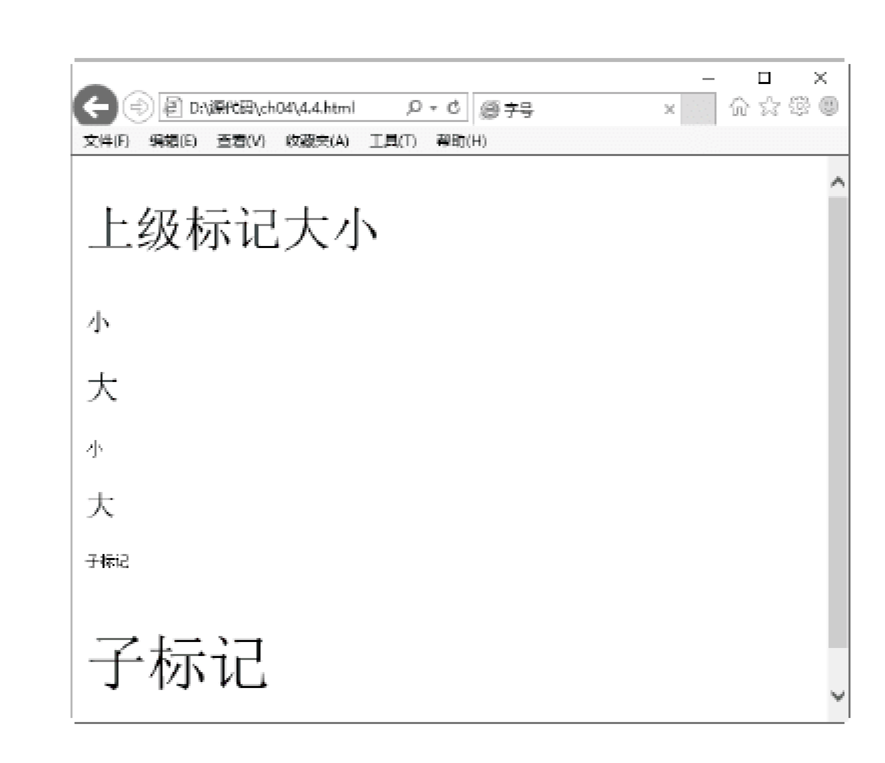


图 4-4　设置文字字号

4.2.3 案例 5——设置文字颜色

没有色彩的网页是枯燥而无生机的，这就意味着一个优秀的网页设计者不仅要能够合理安排页面布局，而且还要具有一定的色彩视觉和色彩搭配能力，这样才能够使网页更加精美也更具表现力，并给浏览者以亲切感。

通常使用 color 属性来设置颜色。其属性值通常使用下面的方式设定，如表 4-2 所示。

表 4-2 颜色设定方式

属 性 值	说 明
color_name	规定颜色值为颜色名称的颜色(如 red)
hex_number	规定颜色值为十六进制值的颜色(如#ff0000)
rgb_number	规定颜色值为 rgb 代码的颜色(如 rgb(255,0,0))
inherit	规定应该从父元素继承颜色
hsl_number	规定颜色值为 HSL 代码的颜色(如 hsl(0,75%,50%))，此为新增加的颜色表现方式
hsla_number	规定颜色值为 HSLA 代码的颜色(如 hsla(120,50%,50%,1))，此为新增加的颜色表现方式
rgba_number	规定颜色值为 RGBA 代码的颜色(如 rgba(125,10,45,0.5))，此为新增加的颜色表现方式

【例 4.5】设置文字颜色(案例文件：ch04\4.5.html)。

```
<!DOCTYPE html>
<html>
<head><title>字体颜色</title>
</head>
<body>
<h1 style="color:#033">页面标题</h1>
<p style="color:red">本段内容用于显示红色。
</p>
<p style="color:rgb(0,0,0)">此处使用 rgb 方式表示了一个黑色文本。</p>
<p style="color:hsl(0,60%,30%)">此处使用新增的 HSL 函数，构建颜色。</p>
<p style="color:hsla(100,50%,50%,1)">此处使用新增加的 HSLA 函数，构建颜色。</p>
<p style="color:rgba(125,20,45,0.5)">此处使用新增加的 RGBA 函数，构建颜色。</p>
</body>
</html>
```

在 IE 10 中预览，效果如图 4-5 所示，可以看到文字以不同颜色显示，并采用了不同的颜色取值方式。



图 4-5 设置文字颜色

4.2.4 案例 6——设置粗体、斜体、下画线

1. 粗体文本

重要文本通常以粗体、强调方式或加强调方式显示。HTML 中的<b>标记、<em>标记和<strong>标记分别实现了这 3 种显示方式。

【例 4.6】重要文本的显示(案例文件：ch04\4.6.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>无标题文档</title>
</head>
<body>
<p><b>我是粗体文字</b> </p>
<p><em>我是强调文字</em> </p>
<p><strong>我是加强调文字</strong></p>
</body>
</html>
```

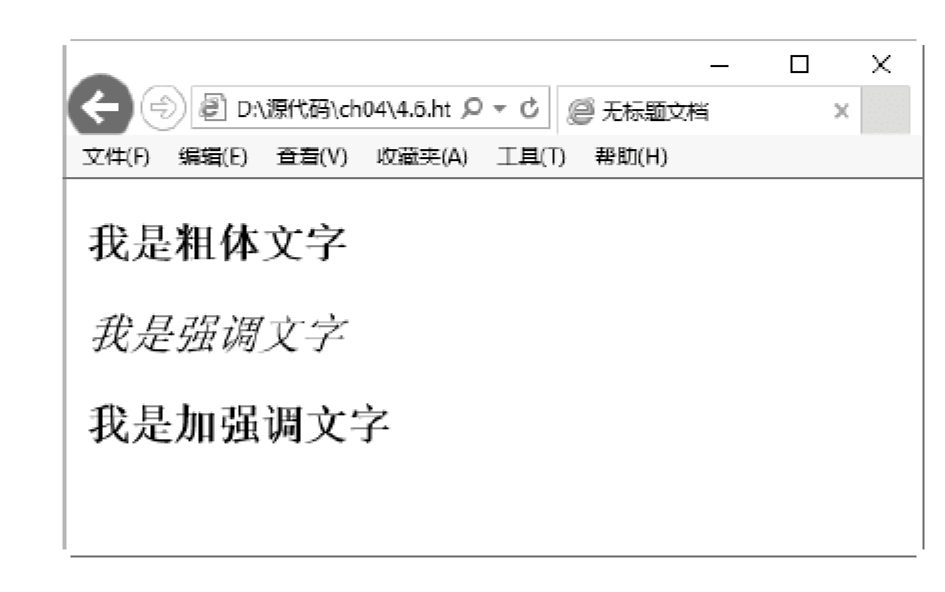


图 4-6 重要文本的显示

在 IE 11.0 中预览，效果如图 4-6 所示，实现了文本的 3 种显示方式。

2. 倾斜文本

HTML 5 中的<i>标记实现了文本的倾斜显示。放在<i></i>之间的文本将以斜体显示。

【例 4.7】设置倾斜文本(案例文件：ch04\4.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>倾斜文本</title>
</head>
<body>
<i>我将会以斜体字显示</i>
</body>
</html>
```

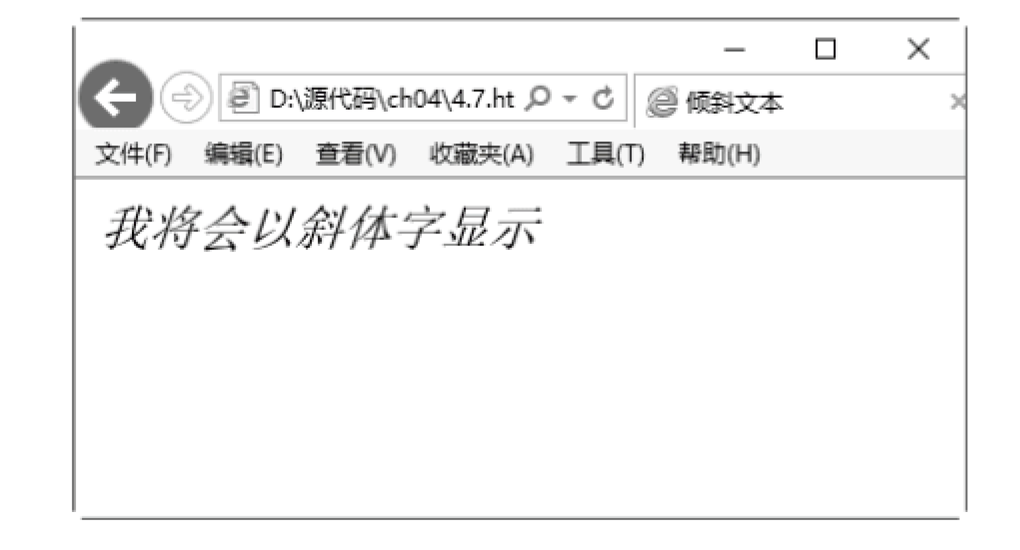
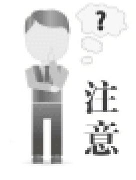


图 4-7 斜体文本的显示

在 IE 11.0 中预览，效果如图 4-7 所示，其中文字以斜体显示。



HTML 中的重要文本和倾斜文本标记已经过时，这些标记都应该使用 CSS 样式来实现。随着后面学习的深入，读者会逐渐发现，即使 HTML 和 CSS 实现相同的效果，但是 CSS 所能实现的控制远远比 HTML 要细致、精确。

3. 为文本添加下画线

HTML 5 中的<u>标记可以为文本添加下画线，放在< u ></ u >之间的文本以添加下画线的方式显示。

【例 4.8】为文本添加下画线(案例文件：ch04\4.8.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>如果文本不是超链接，请尽量不要<u>对其使用下画线</u>。</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-8 所示，其中文字以添加了下画线的方式显示。



请尽量避免为文本添加下画线，因为用户会把它混淆为一个超链接。

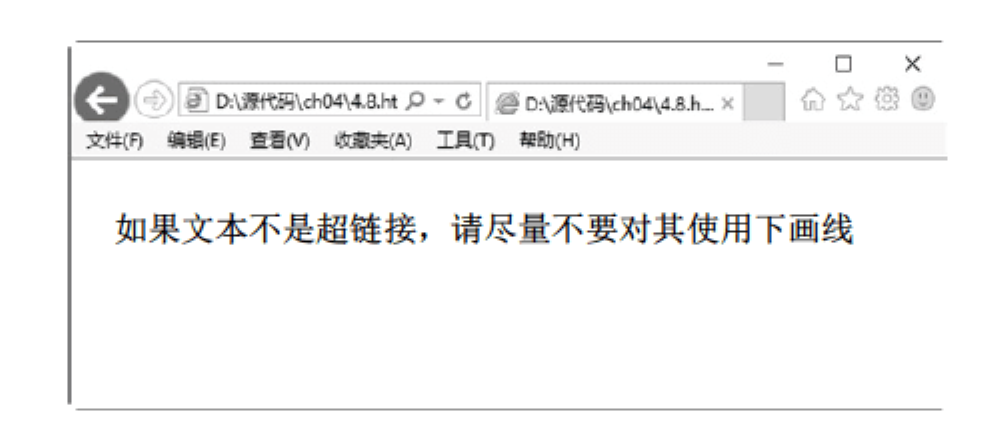


图 4-8　为文本添加下划线

4.2.5　案例 7——设置上标与下标

在 HTML 中用<sup>标记实现上标文字，用<sub>标记实现下标文字。<sup>和<sub>都是双标记，放在开始标记和结束标记之间的文本会分别以上标或下标形式显示。

【例 4.9】设置上标与下标(案例文件：ch04\4.9.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>无标题文档</title>
</head>
<body>
 <!-上标显示-->
 <p>c=a<sup>2</sup>+b<sup>2</sup></p>
<!-下标显示-->
 <p>H<sub>2</sub>+O→H<sub>2</sub>O</p>
</body>
</html>
```

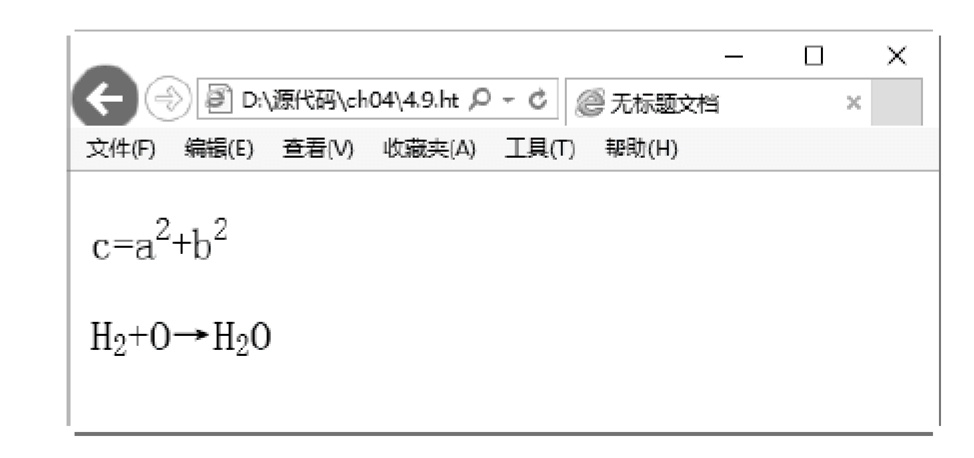


图 4-9　上标和下标预览效果

在 IE 11.0 中预览，效果如图 4-9 所示，分别实现了上标和下标文本显示。

4.2.6　案例 8——设置字体风格

font-style 通常用来定义字体风格，即字体的显示样式。在 HTML 5 新规定中，使用 font-style 的语法格式如下：

```
font-style : normal | italic | oblique |inherit
```

其属性值有 4 个，具体含义如表 4-3 所示。

表 4-3　font-style 的属性值

属 性 值	含　义
normal	默认值。浏览器显示一个标准的字体样式
italic	浏览器会显示一个斜体的字体样式



续表

属 性 值	含　义
oblique	没有斜体变量的特殊字体，浏览器会显示一个倾斜的字体样式
inherit	规定应该从父元素继承字体样式

【例 4.10】使用 font-style 定义字体风格(案例文件：ch04\4.10.html)。

```
<!DOCTYPE html>
<html>
<head><title>字体风格</title></head>
<body>
  <p style="font-style:italic">锄禾日当午，汗滴禾下土</p>
  <p style="font-style:normal">锄禾日当午，汗滴禾下土</p>
  <p style="font-style:oblique">锄禾日当午，汗滴禾下土</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-10 所示，可以看到文字分别显示不同的样式，如斜体。

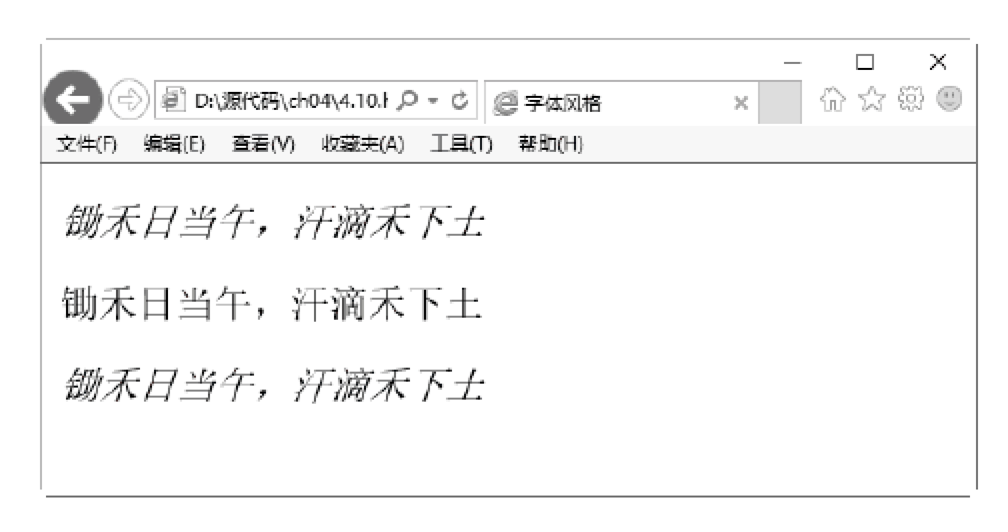


图 4-10　使用 font-style 定义字体风格

4.2.7　案例 9——设置加粗字体

通过设置字体粗细，可以让文字显示不同的外观。通过 font-weight 属性可以定义字体的粗细程度。其语法格式如下。

```
font-weight:100-900|bold|bolder|lighter|normal;
```

font-weight 属性有 13 个有效值，分别是 bold、bolder、lighter、normal、100～900。如果没有设置该属性，则使用其默认值 normal。属性值设置为 100～900，值越大，加粗的程度就越高。font-weight 属性值的具体含义如表 4-4 所示。

表 4-4　font-weight 的属性值

属 性 值	描　述
bold	定义粗体字体
bolder	定义更粗的字体，相对值
lighter	定义更细的字体，相对值
normal	默认，标准字体

浏览器默认的字体粗细是 400，另外也可以通过参数 lighter 和 bolder 使得字体在原有基础上显得更细或更粗。

【例 4.11】加粗字体显示(案例文件：ch04\4.11.html)。

```
<!DOCTYPE html>
<html>
<head><title>加粗字体</title></head>
<body>
   <p style="font-weight:bold">万水千山总是情(bold)</p>
   <p style="font-weight:bolder">万水千山总是情(bolder)</p>
   <p style="font-weight:lighter">万水千山总是情(lighter)</p>
   <p style="font-weight:normal">万水千山总是情(normal)</p>
   <p style="font-weight:100">万水千山总是情(100)</p>
   <p style="font-weight:400">万水千山总是情(400)</p>
   <p style="font-weight:900">万水千山总是情(900)</p>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 4-11 所示，可以看到文字以不同方式加粗，其中使用了关键字加粗和数值加粗。



图 4-11　加粗字体显示

4.2.8　案例 10——设置字体复合属性

在设计网页时，为了使网页布局合理且文本规范，对字体设计需要使用多种属性。例如定义字体粗细，并定义字体大小。但是，多个属性分别书写相对比较麻烦，在 HTML 5 中提供了 font 属性就解决了这一问题。

font 属性可以一次性地使用多个属性的属性值定义文本字体。其语法格式如下：

```
font:font-style font-variant font-weight font-size font-family
```

font 属性中的属性排列顺序是 font-style、font-variant、font-weight、font-size 和 font-family，各属性的属性值之间使用空格隔开。但是，如果 font-family 属性要定义多个属性值，则需要使用逗号(,)隔开。

在属性排列中，font-style、font-variant 和 font-weight 这 3 个属性值是可以自由调换的。而 font-size 和 font-family 则必须按照固定的顺序出现，而且还必须都出现在 font 属性中。如果这两者的顺序不对或缺少一个，那么整条样式规则可能就会被忽略。

【例 4.12】设置字体复合属性(案例文件：ch04\4.12.html)。

```
<!DOCTYPE html>
<html>
<head><title>字体复合属性</title>
<style type=text/css>
p{
   font:normal small-caps bolder 25pt
"Cambria","Times New Roman",黑体
}
</style>
</head>
<body>
<p>
学习 HTML 5 标记语言，开发完美绚丽网站。
</p>
</body>
</html>
```

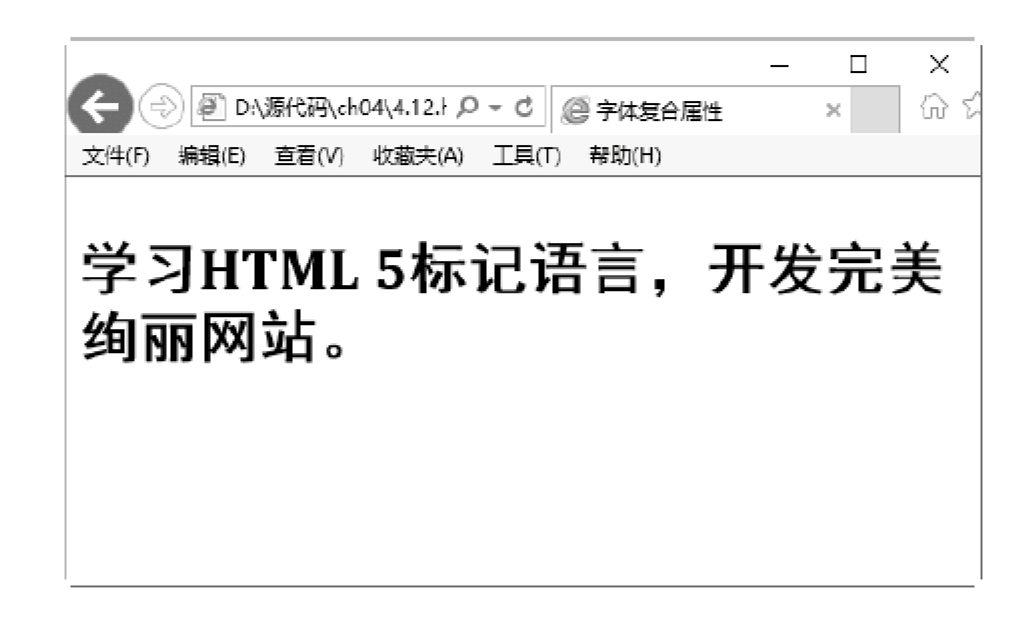


图 4-12 字体复合属性

在 IE 11.0 中浏览，效果如图 4-12 所示，可以看到文字被设置成宋体并加粗。

4.2.9 案例 11——设置阴影文本

在显示字体时，有时根据需求，需要给出文字的阴影效果，以增强网页整体的吸引力，并且为文字阴影添加颜色。这时就需要用到 text-shadow 属性，其语法格式如下：

```
text-shadow : none | <length> none | [<shadow>, ] * <opacity> 或 none |
<color> [, <color> ]*
```

text-shadow 的属性值如表 4-5 所示。

表 4-5 text-shadow 的属性值

属性值	说 明
<color>	指定颜色
<length>	由浮点数字和单位标识符组成的长度值。可为负值。指定阴影的水平延伸距离
<opacity>	由浮点数字和单位标识符组成的长度值。不可为负值。指定模糊效果的作用距离。如果仅仅需要模糊效果，则将前两个 length 全部设定为 0

text-shadow 属性有 4 个值，最后两个值是可选的。第一个属性值表示阴影的水平偏移，可取正负值；第二个值表示阴影垂直偏移，可取正负值；第三个值表示阴影模糊半径，该值可选；第四个值表示阴影颜色值，该值可选。其语法格式如下：

```
text-shadow:阴影水平偏移值(可取正负值)； 阴影垂直偏移值(可取正负值)；阴影模糊值；阴影颜色
```

【例 4.13】设置阴影文本(案例文件：ch04\4.13.html)。

```
<!DOCTYPE html>
<html>
```

```
<head><title>阴影文本</title>
</head>
<body>
<p align=center style="text-
shadow:0.1em 3px 6px blue;font-
size:80px;"> 春蚕到死丝方尽</br>
蜡炬成灰泪始干</p>
</body>
</html>
```

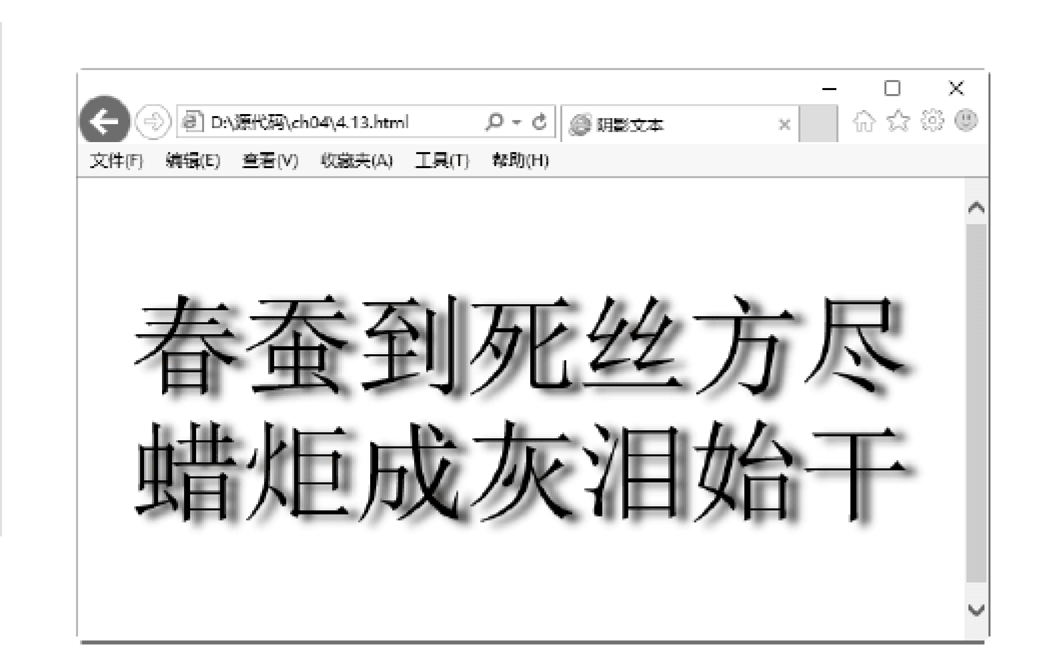
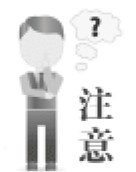


图 4-13　文字居中并阴影显示

在 IE 11.0 中浏览，效果如图 4-13 所示，可以看到文字居中并带有阴影显示。

通过上述实例，可以看出阴影偏移由两个 length 值指定到文本的距离。第一个长度值指定到文本右边的水平距离，负值会把阴影放置在文本左边。第二个长度值指定到文本下边的垂直距离，负值会把阴影放置在文本上方。在阴影偏移之后，可以指定一个模糊半径。



注意

模糊半径是一个长度值，它指定了模糊效果的范围，但如何计算效果的具体算法，并没有指定。在阴影效果的长度值之前或之后，还可以指定一个颜色值。颜色值会被用作阴影效果的基础。如果没有指定颜色，那么将使用 color 属性值来替代。

4.2.10　案例 12——控制换行

当在一个指定区域显示一整行文字时，如果文字在一行显示不完时，则需要进行换行。如果不进行换行，则会超出指定区域范围，此时我们可以采用新增加的 word-wrap 文本样式，来控制文本换行。

word-wrap 的语法格式如下：

```
word-wrap :  normal | break-word
```

其属性值含义比较简单，如表 4-6 所示。

表 4-6　word-wrap 的属性值

属 性 值	说　明
normal	控制连续文本换行
break-word	内容将在边界内换行。如果需要，词内换行(word-break)也会发生

【例 4.14】控制文本换行(案例文件：ch04\4.14.html)。

```
<!DOCTYPE html>
<html>
<head><title>控制换行</title></head>
<body>
<style type="text/css">
 div{ width:300px;word-wrap:break-word;border:1px solid #999999;}
</style>
    <div>本文测试控制换行功能，可以使文本在指定框架中换行显示内容。</div><br>

<div>wordwrapbreakwordwordwrapbreakwordwordwrapbreakwordwordwrapbreakword</
```

```
div><br>
   <div>This is all English,This is all
English,This is all English,This is all
English,</div>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 4-14 所示，可以看到文字在指定位置被控制换行。

可以看出，word-wrap 属性可以控制换行，当属性取值 break-word 时，将强制换行，中文文本没有任何问题，英文语句也没有任何问题。但是对于长串的英文则不起作用，也就是说，break-word 属性是控制是否断词，而不是断字符。

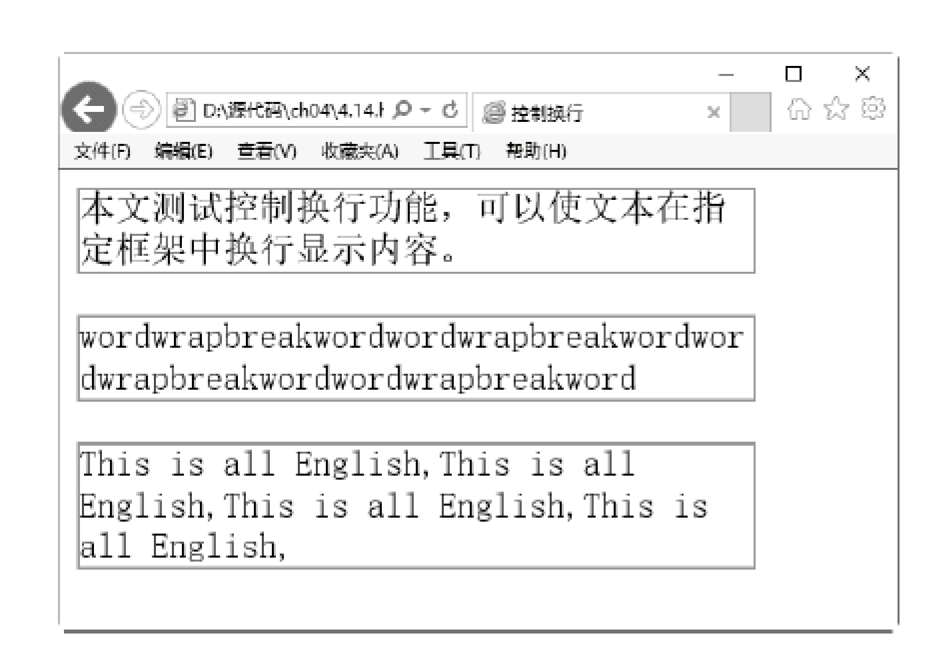


图 4-14　控制文本换行

4.3　设置段落格式

在网页中如果要把文字合理地显示出来，离不开段落标记的使用。对网页中文字段落进行排版，并不像文本编辑软件 Word 那样可以定义许多模式来安排文字的位置。在网页中要让某一段文字放在特定的地方是通过 HTML 标记来完成的。

4.3.1　案例 13——设置段落标记

段落标记是双标记，即<p></p>，在<p>开始标记和</p>结束标记之间的内容形成一个段落。如果省略结束标记，从<p>标记开始，直到遇见下一个段落标记之前的文本，都在一个段落内。段落标记中的 p 是英文单词 paragraph 即“段落”的首字母，用来定义网页中的一段文本，文本在一个段落中会自动换行。

【例 4.15】设置段落标记(案例文件：ch04\4.15.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>段落标记的使用</title>
</head>
<body>
 <p>HTML 5、CSS3 应用教程之 跟 DIV 说 Bye!Bye!</p>
<p>Web 设计师可以使用 HTML4 和 CSS2.1 完成一些很酷的东西。我们可以在不使用陈旧的基于
table 布局的基础上完成文档逻辑结构并创建内容丰富的网站。我们可以在不使用内联<font>和
<br>标签的基础上对网站添加漂亮而细腻的风格样式。事实上，我们目前的设计能力已经让我们远离
了那个可怕的浏览器战争时代、专有协议和那些充满闪动、滚动和闪烁的丑陋网页。
<p>
<p>
虽然我们现在已经普遍使用了 HTML4 和 CSS2.1，但是我们还可以做得更好！我们可以重组我们代码
的结构并能让我们的页面代码更富有语义化特性。我们可以缩减带给页面美丽外观样式代码量并让它们
有更高的可扩展性。现在，HTML 5 和 CSS3 正跃跃欲试地等待大家，下面让我们来看看它们是否真的
能让我们的设计提升到下一个高度吧…
</p>
<p>
曾经，设计师们经常会更频繁使用基于 table 的没有任何语义的布局。不过最终还是要感谢像
```

```
Jeffrey Zeldman 和 Eric Meyer 这样的思想革新者，聪明的设计师们慢慢地接受了相对更语义化
的<div>布局替代了 table 布局，并且开始调用外部样式表。但不幸的是，复杂的网页设计需要大量
不同的标签结构代码，我们把它叫作“<div>-soup” 综合征。
</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-15 所示，<P>标记将文本分成 4 个段落。

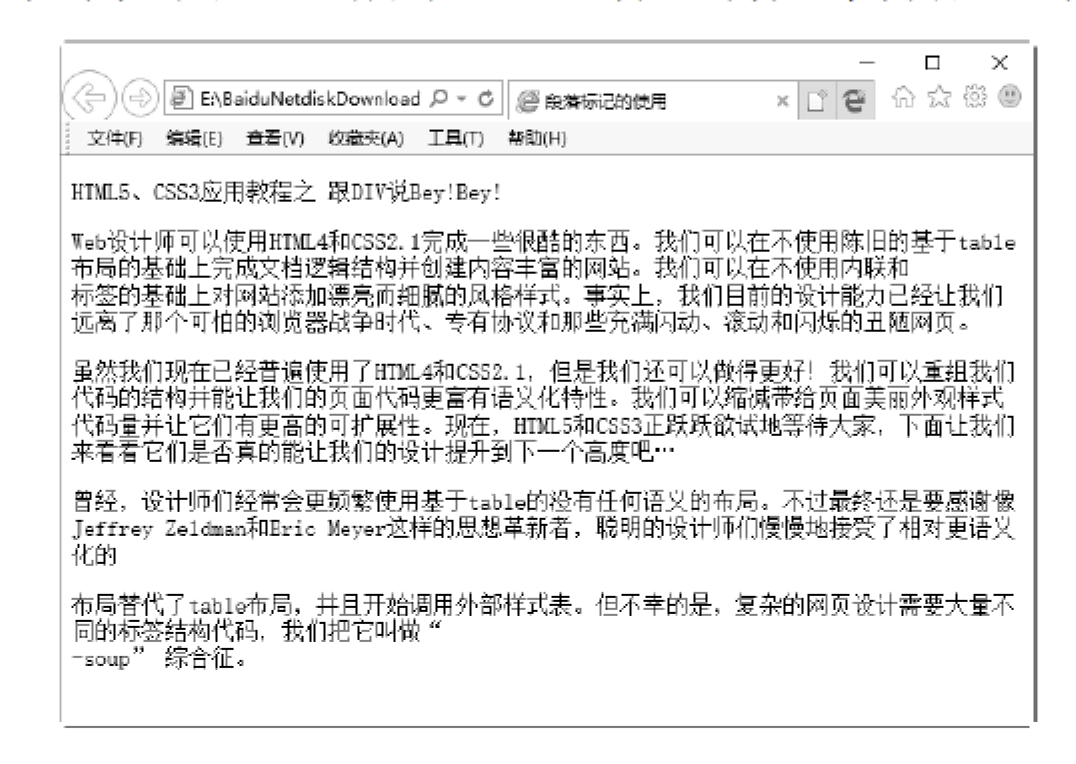


图 4-15　段落标记的使用

4.3.2　案例 14——设置换行标记

换行标记
是一个单标记，它没有结束标记，是英文单词 break 的缩写，作用是将文字在一个段内强制换行。一个
标记代表一个换行，连续的多个标记可以实现多次换行。使用换行标记时，在需要换行的位置添加
标记即可。例如，下面的代码实现了对文本的强制换行。

【例 4.16】设置换行标记(案例文件：ch04\4.16.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文本段换行</title>
</head>
<body>
本节目标<br />
网页中的文字是如何设置的<br/>
如何在 Dreamweaver 中处理文字<br/>
如何对文本进行格式化(CSS)<br />
熟悉使用 Dreamweaver 进行样式表的创建与应用
</body>
</html>
```

虽然在 HTML 源代码中，主体部分的内容在排版上没有换行，但是增加
标记后，在 IE 11.0 中预览，效果如图 4-16 所示，实现了换行效果。

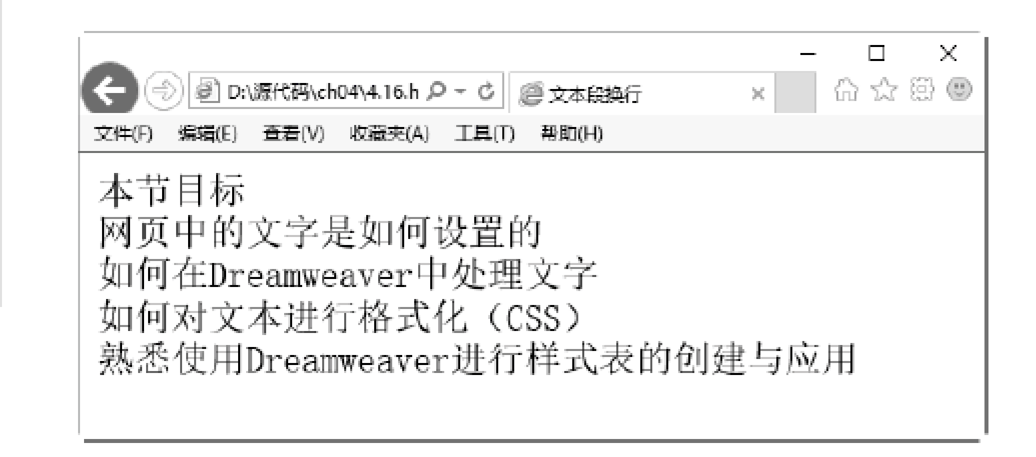


图 4-16　换行标记的使用

4.4 设置网页水平线

使用<hr>标签可以在 HTML 页面中创建一条水平线，并设置水平线的高度、宽度、颜色、对齐方式等样式。

4.4.1 案例 15——添加水平线

在 HTML 中，<hr>标签没有结束标签。

【例 4.17】添加水平线(案例文件：ch04\4.17.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>hr 标签定义水平线</p>
<hr />
<p>这是一个段落。</p>
<hr />
<p>这是一个段落。</p>
<hr />
<p>这是一个段落。</p>
</body>
</html>
```

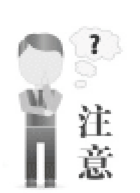


图 4-17 添加水平线

在 IE 11.0 中预览，效果如图 4-17 所示，在其中可以看到添加的水平线。

4.4.2 案例 16——设置水平线的宽度与高度

使用 size 与 width 属性可以设置水平线的宽度与高度。其中，width 属性规定水平线的宽度，以像素计或百分比计；size 属性规定水平线的高度，以像素计。



HTML 5 已经不再支持<hr>标记的 size 与 width 属性。

【例 4.18】设置水平线的宽度与高度(案例文件：ch04\4.18.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>普通的水平线</p>
<hr />
<p>高度为 50 像素的水平线</p>
```

```
<hr size="50" />
<p>宽度为 50%的水平线</p>
<hr width="30%" />
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-18 所示，其中一条水平线的高度为 50 像素，一条水平线的宽度为 50%。

图 4-18　设置水平线的宽度与高度

4.4.3　案例 17——设置水平线的颜色

使用 color 属性可以设置水平线的颜色。下面以给网页添加一个红色水平线为例，来介绍设置水平线颜色的方法。

【例 4.19】设置水平线的颜色(案例文件：ch04\4.19.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>下面是一条红色水平线</p>
<hr color="red" />
</body>
</html>
```

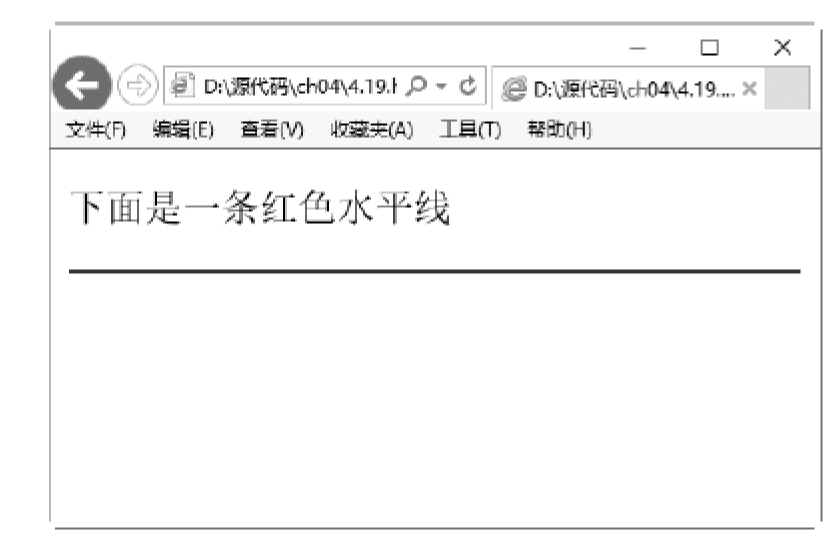


图 4-19　设置水平线的颜色

在 IE 11.0 中预览，效果如图 4-19 所示，可以看出网页中显示的水平线是红色。

4.4.4　案例 18——设置水平线的对齐方式

align 属性规定水平线的水平对齐方式，包括 3 种对齐方式，分别是：left(左对齐)、right(右对齐)、center(居中对齐)。需要提示用户的是除非 width 属性设置为小于 100%，否则 align 属性不会有任何效果。



HTML 5 已经不再支持<hr>标记的 align 属性。

【例 4.20】设置水平线的对齐方式(案例文件：ch04\4.20.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>设置水平线的对齐方式</p>
<hr align="center" width="50%" />
<hr align="left" width="50%" />
<hr align="right" width="50%" />
</body>
</html>
```



图 4-20　设置水平线的对齐方式

在 IE 11.0 中预览，效果如图 4-20 所示，可以看到



网页中水平线的对齐方式。

4.4.5 案例 19——去掉水平线阴影

noshade 属性规定水平线的颜色呈现为纯色，而不是有阴影的颜色。下面介绍去掉水平线阴影的方法。



HTML 5 已经不再支持<hr>标记的 noshade 属性。

【例 4.21】去掉水平线阴影(案例文件：ch04\4.21.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>带阴影的水平线(默认)</p>
<hr />
<p>不带阴影的水平线</p>
<hr noshade="noshade" />
</body>
</html>
```

在 IE 11.0 中预览，效果如图 4-21 所示，可以看到两种水平线的区别。



图 4-21 去掉水平线的阴影

4.5 综合案例——成才教育网文本设计

本章讲述了网页组成元素中最常用的文本。本实例将综合运用网页文本的设计方法，制作成才教育网的文本页面。其具体操作步骤如下。

step 01 在 Dreamweaver CC 中新建 HTML 文档，并修改成 HTML 5 标准。代码如下：

```
<!DOCTYPE html>
<html >
<head>
<meta charset="utf-8" />
<title>成才教育网</title>
</head>
<body>
</body>
</html>
```

step 02 在 body 部分增加如下 HTML 代码。保存页面。

```
<p><h2>成才教育</h2></p>
<p>成才教育成立于 2003 年，是一家专业致力于学生学习能力开发和培养、学习社区建设、课外辅导服务、家庭教育研究的新型综合教育服务机构。自成立起，一直专业致力于初高中学生的课外辅导和学习能力的培养。</p>
<h3>教学模式</h3>
```

```
<p>为学生量身定制最佳的学习方案，改善学习方法，充分挖掘学生们的智力潜能，激发学习兴趣，培养学生的自学能力，辅导老师(以一线重点在校教师为主)对学生设计适合学生的辅导教案与作业习题。</p>
<h3>教学特色</h3>
    分析学科不足制订辅导计划；<br />
    特级名师高考难点点睛；<br />
    专人陪读随时解除疑难；<br />
    专业学科导师一对一面授学科知识、解题技巧、学习方式。
```

step 03 使用 IE 打开文件预览，效果如图 4-22 所示。



图 4-22　成才教育网

4.6　高 手 解 惑

疑问 1：换行标记和段落标记的区别？

答：换行标记是单标记，不能写结束标记。段落标记是双标记，可以省略结束标记也可以不省略。在默认情况下，段落之间的距离和段落内部的行间距是不同的，段落间距比较大，行间距比较小。HTML 无法调整段落间距和行间距，如果希望调整它们，就必须使用 CSS。在 Dreamweaver CC 的设计视图下，按 Enter 键可以快速换段，按 Shift+Enter 组合键可以快速换行。

疑问 2：HTML 文档页面上边总是留出一段空白？

答：body 默认有个上边距，设置这个值的属性 topmargin=0 就可以了。有时还需要设置 leftmargin、rightmargin 和 bottommargin 属性值。

第 5 章

设计网页列表与段落

网页列表与段落是网页中最主要也是最常用的元素。其中，网页列表可以有序地编排一些信息资源，使其结构化和条理化，并以列表的样式显示出来，以便浏览者能更加快捷地获得相应信息。网页中用来表达同一个意思的多个文字组合，可以称为段落。段落是文章的基本单位，同样也是网页的基本单位。

重点案例效果



5.1　网页文字列表的设计

HTML 网页中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。下面介绍如何在网页中设计文字列表。

5.1.1　案例 1——建立无序列表

无序列表相当于 Word 中的项目符号。无序列表的项目排列没有顺序，只以符号作为分项标识。无序列表使用一对标记
，其中每一个列表项使用
，其结构如下：

```
<ul>
  <li>无序列表项</li>
  <li>无序列表项</li>
  <li>无序列表项</li>
  <li>无序列表项</li>
</ul>
```

在无序列表结构中，使用
标记表示这一个无序列表的开始和结束；<li>则表示一个列表项的开始。在一个无序列表中可以包含多个列表项，并且<li>可以省略结束标记。下面的实例使用无序列表实现文本的排列显示。

【例 5.1】建立无序列表(案例文件：ch05\5.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>嵌套无序列表的使用</title>
</head>
<body>
<h1>网站建设流程</h1>
<ul>
    <li>项目需求</li>
    <li> 系统分析
      <ul>
        <li>网站的定位</li>
        <li>内容收集</li>
        <li>栏目规划</li>
        <li>网站目录结构设计</li>
        <li>网站标志设计</li>
        <li> 网站风格设计</li>
        <li> 网站导航系统设计</li>
      </ul>
   </li>
   <li> 伪网页草图
      <ul>
        <li> 制作网页草图</li>
        <li>将草图转换为网页</li>
      </ul>
   </li>
   <li> 站点建设</li>
```

```
    <li>网页布局</li>
    <li> 网站测试</li>
    <li> 站点的发布与站点管理 </li>
</ul>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 5-1 所示。显然，无序列表项中，可以嵌套一个列表。例如，代码中的“系统分析”列表项和“伪网页草图”列表项中都有下级列表，因此在这对
标记间又增加了一对
标记。

网站建设流程

- 项目需求
- 系统分析
  - 网站的定位
  - 内容收集
  - 栏目规划
  - 网站目录结构设计
  - 网站标志设计
  - 网站风格设计
  - 网站导航系统设计
- 伪网页草图
  - 制作网页草图
  - 将草图转换为网页
- 站点建设
- 网页布局
- 网站测试
- 站点的发布与站点管理

图 5-1　无序列表

5.1.2　案例 2——建立有序列表

有序列表类似于 Word 中的自动编号功能。有序列表的使用方法和无序列表的使用方法基本相同。它使用标记
，每一个列表项前使用
。每个项目都有前后顺序之分，多数用数字表示，其结构如下：

```
<ol>
  <li>第 1 项</li>
  <li>第 2 项</li>
  <li>第 3 项</li>
</ol>
```

下面的实例使用有序列表实现文本的排列显示。

【例 5.2】建立有序列表(案例文件：ch05\5.2.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>有序列表的使用</title>
</head>
<body>
<h1>本讲目标</h1>
<ol>
  <li>网页的相关概念 </li>
  <li> 网页与 HTML</li>
  <li> Web 标准(结构、表现、行为)</li>
  <li> 网页设计与开发的过程  </li>
  <li>与设计相关的技术因素</li>
  <li> HTML 简介 </li>
</ol>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 5-2 所示。可以看到新添加的有序列表。



图 5-2　有序列表

5.1.3　案例 3——建立不同类型的无序列表

通过使用多个<ul>标签，可以建立不同类型的无序列表。

【例 5.3】建立不同类型的无序列表(案例文件：ch05\5.3.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>Disc 项目符号列表：</h4>
<ul type="disc">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>
<h4>Circle 项目符号列表：</h4>
<ul type="circle">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>
<h4>Square 项目符号列表：</h4>
<ul type="square">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 5-3 所示。可以看到，网页中插入了 3 种不同类型的无序列表。

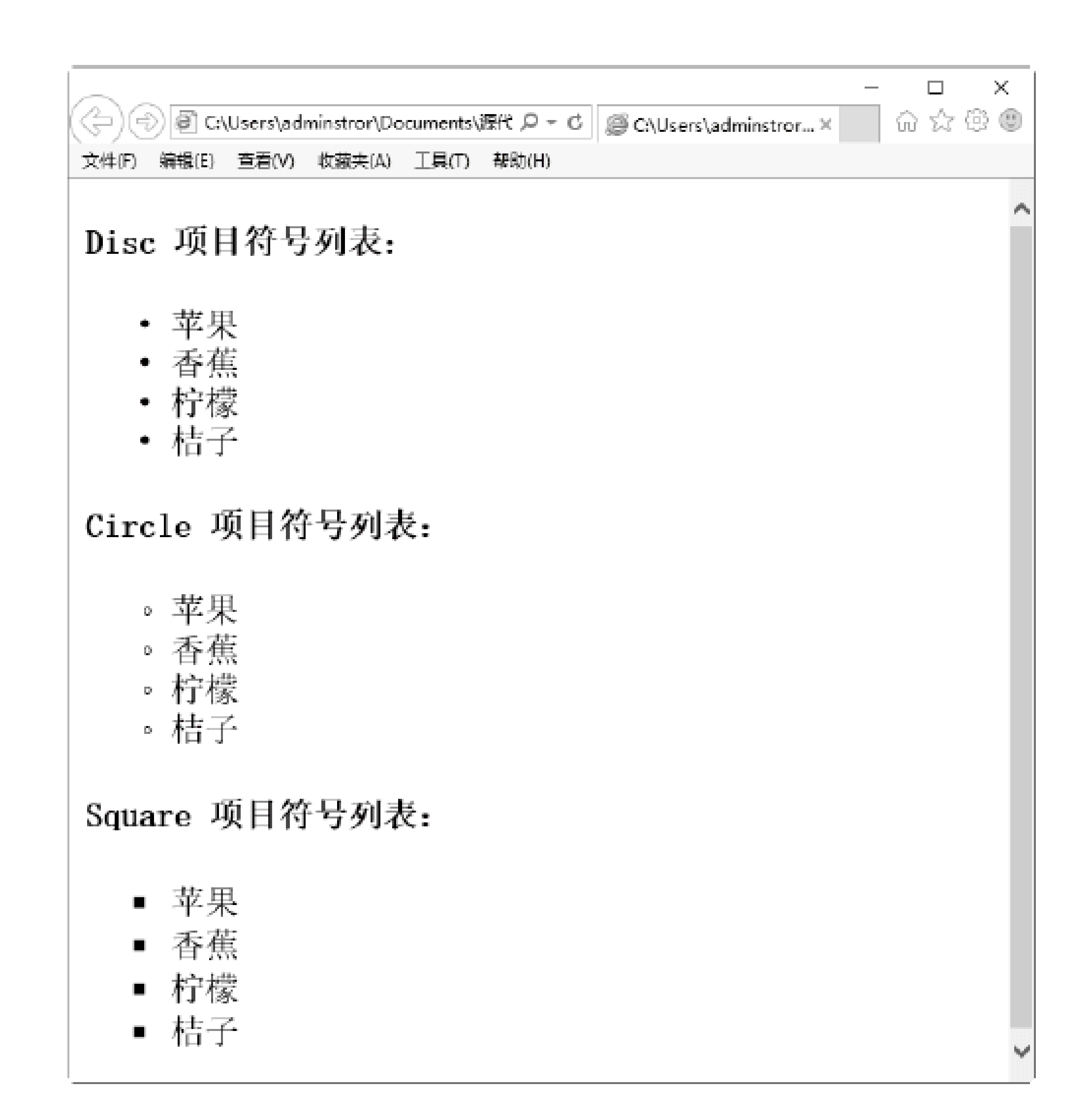


图 5-3　不同类型的无序列表

5.1.4　案例 4——建立不同类型的有序列表

通过使用多个<ol>标签，可以建立不同类型的有序列表。

【例 5.4】建立不同类型的有序列表(案例文件：ch05\5.4.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>数字列表：</h4>
<ol>
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>
```

```
<h4>字母列表：</h4>
<ol type="A">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 5-4 所示。可以看到，网页中建立了两种不同类型的有序列表。

图 5-4 不同类型的有序列表

5.1.5 案例 5——嵌套列表

嵌套列表是网页中常用的元素，使用<ul>标签可以制作网页中的嵌套列表。

【例 5.5】建立网页嵌套列表(案例文件：ch05\5.5.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>一个嵌套列表：</h4>
<ul>
  <li>咖啡</li>
  <li>茶
    <ul>
    <li>红茶</li>
    <li>绿茶
      <ul>
      <li>中国茶</li>
      <li>非洲茶</li>
      </ul>
    </li>
    </ul>
  </li>
  <li>牛奶</li>
</ul>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 5-5 所示。可以看到，网页中插入了一个嵌套列表。



图 5-5 网页嵌套列表

5.1.6 案例 6——自定义列表<dl>

在 HTML 5 中还可以自定义列表，所用的标签是<dl>。

【例 5.6】设计自定义网页列表(案例文件：ch05\5.6.html)。

```
<!DOCTYPE html>
<html>
<body>
<h2>一个定义列表：</h2>
```

```
<dl>
   <dt>电脑</dt>
   <dd>是一种能够按照程序运行的电子设备…</dd>
   <dt>显示器</dt>
   <dd>以视觉方式显示信息的装置…</dd>
</dl>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 5-6 所示。可以看到，网页中建立了一个自定义列表。



图 5-6　自定义列表

5.2　网页段落格式的设计

段落的放置与效果的显示会直接影响到页面的布局及风格。在 HTML 5 中有关文本段落的格式设置需要靠 CSS 样式来实现。CSS 样式表提供了文本属性来实现对页面中段落文本的控制。

5.2.1　案例 7——设计单词间隔 word-spacing

单词之间的间隔如果设置合理，一是会给整个网页布局节省空间；二是可以给人赏心悦目的感觉，提高阅读效果，在 CSS 中，可以使用 word-spacing 直接定义指定区域或者段落中字符之间的间隔。

word-spacing 属性用于设定词与词之间的间距，即增加或者减少词与词之间的间隔。其语法格式如下：

```
word-spacing : normal | length
```

其中属性值 normal 和 length 的含义如表 5-1 所示。

表 5-1　word-spacing 的属性值

属 性 值	说　明
normal	默认，定义单词之间的标准间隔
length	定义单词之间的固定宽度，可以接受正值或负值

【例 5.7】定义段落中单词的间隔(案例文件：ch05\5.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>单词间隔</title>
</head>
<body>
<p style="word-spacing:normal">Welcome to Beijing!</p>
<p style="word-spacing:10px">Welcome to Beijing!</p>
<p style="word-spacing:10px">北京欢迎您!</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-7 所示。可以看到，段落中单词以不同间隔显示。

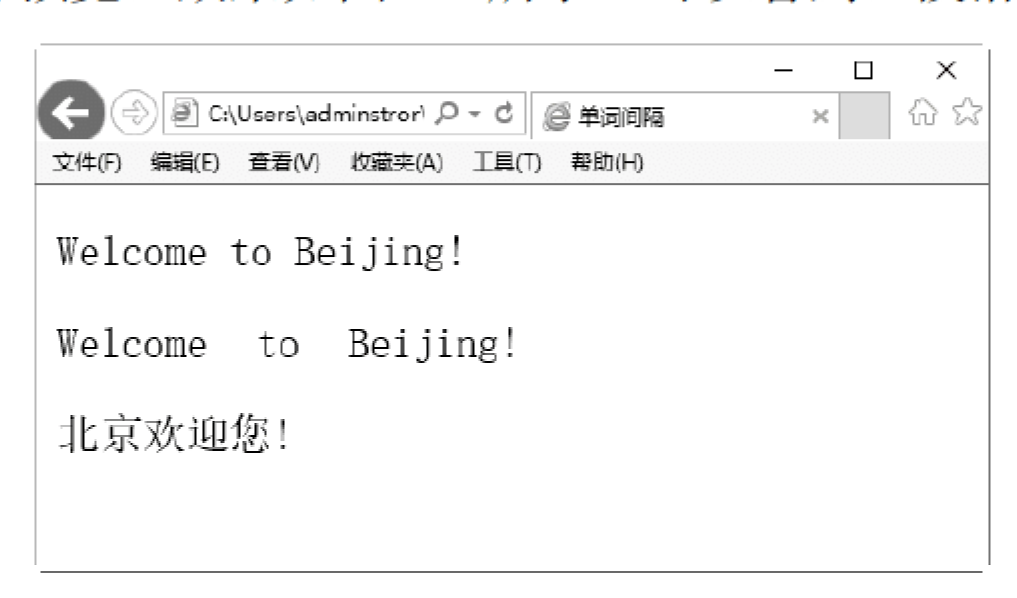


图 5-7　定义单词的间隔



从上面的显示结果可以看出，word-spacing 属性不能用于设定文字之间的间隔。

5.2.2　案例 8——设计字符间隔 letter-spacing

在一个网页中，还可能涉及多个文本字符，将文本字符之间的间距，设置得和词间隔保持一致，进而保持页面的整体性，是网页设计者必须完成的。词与词之间可以通过 word-spacing 进行设置，那么字符之间使用什么设置呢？

通过 letter-spacing 样式可以设置文本字符之间的距离。即在文本字符之间插入多少空间，这里允许使用负值，这会让字母之间更加紧凑。其语法格式如下：

```
letter-spacing : normal | length
```

letter-spacing 的属性值含义如表 5-2 所示。

表 5-2　letter-spacing 的属性值

属 性 值	说　明
normal	默认间隔，即以字符之间的标准间隔显示
length	由浮点数字和单位标识符组成的长度值，允许为负值

【例 5.8】定义段落中字符的间隔(案例文件：ch05\5.8.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>字符间隔</title>
</head>
<body>
<p style="letter-spacing:normal">Welcome to Beijing!</p>
<p style="letter-spacing:10px">Welcome to Beijing!</p>
<p style="letter-spacing:2ex">设置字符间距为 2ex</p>
<p style="letter-spacing:-0.5ex">设置字符间距为-0.5ex</p>
<p style="letter-spacing:2em">设置字符间距为 2em</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-8 所示，可以看到，文字间距以不同大小显示。



图 5-8　定义字符的间隔



从上述代码中可以看出，通过 letter-spacing 定义了多个字间距的效果。特别注意，当设置的字间距是-0.5ex 时，文字就会拥挤到一块。

5.2.3　案例 9——设计文字修饰 text-decoration

在网页文本编辑中有的文字需要突出重点，即告诉读者这段文本的重要作用，这时往往会给其增加下画线，或者增加上画线和删除线效果，从而吸引读者的眼球。可以使用 text-decoration 文本修饰属性为页面提供多种文本的修饰效果，如下画线、删除线、闪烁等。其语法格式如下：

```
text-decoration:none||underline||blink||overline||line-through
```

text-decoration 的属性值含义如表 5-3 所示。

表 5-3　text-decoration 的属性值

属 性 值	描　述
none	默认值，对文本不进行任何修饰
underline	下画线
overline	上画线
line-through	删除线
blink	闪烁

【例 5.9】设计段落中文字的修饰效果(案例文件：ch05\5.9.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文字修饰</title>
</head>
<body>
<p style="text-decoration:none">悠悠中华五千年!</p>
<p style="text-decoration:underline">悠悠中华五千年!</p>
<p style="text-decoration:overline">悠悠中华五千年!</p>
```

```
<p style="text-decoration:line-through">悠悠中华五千年!</p>
<p style="text-decoration:blink">悠悠中华五千年!</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-9 所示，可以看到，段落中出现了下画线、上画线、删除线等。



这里需要注意的是：blink 闪烁效果只有 Mozilla 和 Netscape 浏览器支持，而 IE 和其他浏览器(如 Opera)暂不支持该效果。

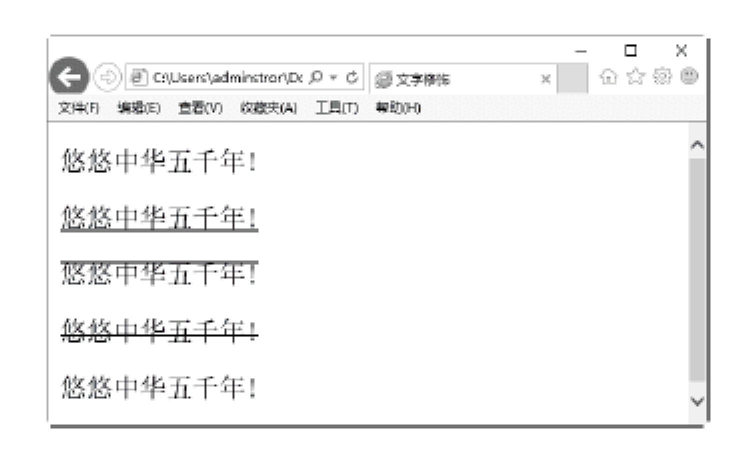


图 5-9 设计文字的修饰效果

5.2.4 案例 10——设计垂直对齐方式 vertical-align

在网页文本编辑中，对齐有很多方式，字符排在一行的中央位置叫“居中”。文章的标题和表格中的数据一般都居中排。有时还要求文字垂直对齐，即文字顶部对齐，或者底部对齐。

在 CSS 中，可以直接使用 vertical-align 属性来定义，该属性用来设定垂直对齐方式。该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐，允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

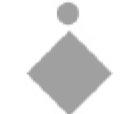
vertical-align 属性的语法格式如下：

```
{vertical-align:属性值}
```

vertical-align 属性值有 8 个预设值可以使用，也可以使用百分比。这 8 个预设值如表 5-4 所示。

表 5-4 vertical-align 的属性值

属 性 值	说 明
baseline	默认。元素放置在父元素的基线上
sub	垂直对齐文本的下标
super	垂直对齐文本的上标
top	把元素的顶端与行中最高元素的顶端对齐
text-top	把元素的顶端与父元素字体的顶端对齐
middle	把此元素放置在父元素的中部
bottom	把元素的顶端与行中最低的元素的顶端对齐
text-bottom	把元素的底端与父元素字体的底端对齐
length	设置元素的堆叠顺序
%	使用 line-height 属性的百分比值来排列此元素。允许使用负值



【例 5.10】设计段落中文字的垂直对齐方式(案例文件：ch05\5.10.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文字修饰</title>
</head>
<body>
<p>
   中国<b style=" font-size:8pt;vertical-align:super">2012</b>神农架<b
style="font-size: 8pt;vertical-align: sub">[注]</b>是一个充满神奇色彩的美丽的地
方！！
   <img src="images/1.gif" style="vertical-align: baseline">
</p>
<p>
   中国<b style=" font-size:8pt;vertical-align:100%">2012</b>万里长城<b
style="font-size: 8pt;vertical-align: -100%">[注]</b>是雄伟壮观的历史遗迹！！
   <img src="images/2.gif" style="vertical-align:middle"/>
   <img src="images/2.gif" style="vertical-align:text-top">
   <img src="images/2.gif" style="vertical-align:bottom">
   <img src="images/2.gif" style="vertical-align:text-bottom">
</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-10 所示。可以看到，文字在垂直方向以不同的对齐方式显示。

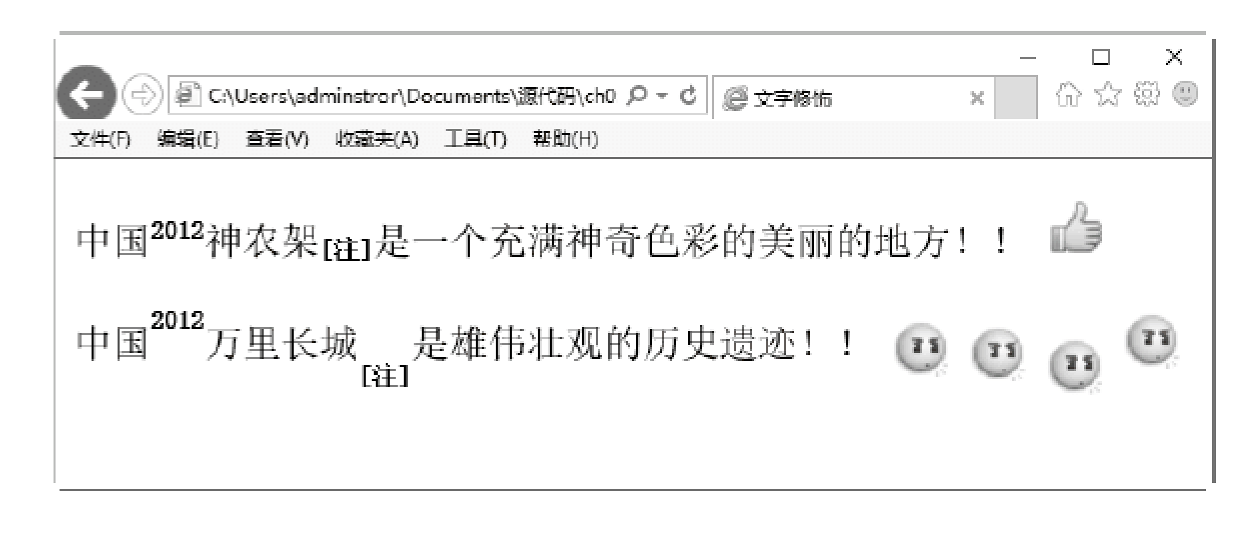


图 5-10　设计文字的垂直对齐方式

从上面实例中，可以看出在页面中的数学运算或注释标号使用上下标比较多。顶端对齐有两种参照方式：一种是参照整个文本块；另一种是参照文本。底部对齐同顶端对齐方式相同，分别参照文本块和文本块中包含的文本。



vertical-align 属性值还能使用百分比来设定垂直高度。该高度具有相对性，它是基于行高的值来计算的。而且百分比还能使用正负号，正百分比使文本上升，负百分比使文本下降。

5.2.5　案例 11——设计文本转换 text-transform

根据需要，将小写字母转换为大写字母，或者将大写字母转换为小写字母，在文本编辑中都是很常见的。使用 text-transform 属性可用于设定文本字体的大小写转换。

text-transform 属性的语法格式如下：

```
text-transform : none | capitalize | uppercase | lowercase
```

text-transform 属性值的含义如表 5-5 所示。

表 5-5 text-transform 的属性值

属 性 值	说 明
none	无转换发生
capitalize	将每个单词的第一个字母转换成大写，其余无转换发生
uppercase	转换成大写
lowercase	转换成小写

因为文本转换属性仅作用于字母型文本，相对来说比较简单。

【例 5.11】设计段落中文本的转换效果(案例文件：ch05\5.11.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文字转换</title>
</head>
<body>
<body>
  <p style="text-transform:none">we are good friends</p>
  <p style="text-transform:capitalize">we are good friends</p>
  <p style="text-transform:uppercase">we are good friends</p>
  <p style="text-transform:lowercase">WE ARE GOOD FRIENDS</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-11 所示。可以看到，字母依照设置显示大小写。

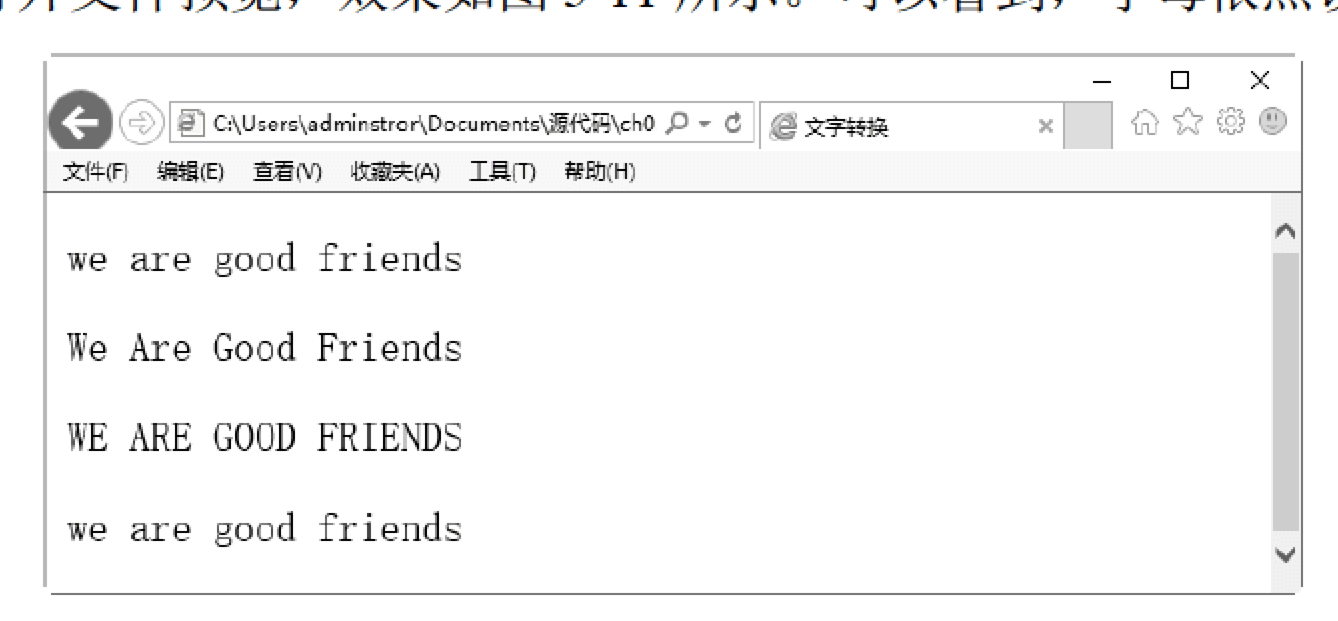


图 5-11 文本的转换效果

5.2.6 案例 12——设计水平对齐方式 text-align

一般情况下，居中对齐适用于标题类文本，其他对齐方式可以根据页面布局来选择使用。根据需要，可以设置多种对齐，如水平方向上的居中、左对齐、右对齐或者两端对齐等。可以通过 text-align 属性完成水平对齐设置。

text-align 属性用于定义文本对象的对齐方式，其语法格式如下：

```
{text-align: 属性值 }
```

text-align 属性值的含义如表 5-6 所示。

表 5-6　text-align 的属性值

属 性 值	说　明
start	文本向行的开始边缘对齐
end	文本向行的结束边缘对齐
left	文本向行的左边缘对齐。在垂直方向的文本中，文本在 left-to-right 模式下向开始边缘对齐
right	文本向行的右边缘对齐。在垂直方向的文本中，文本在 left-to-right 模式下向结束边缘对齐
center	文本在行内居中对齐
justify	文本根据 text-justify 的属性设置方法分散对齐。即两端对齐，均匀分布
match-parent	继承父元素的对齐方式，但有个例外：继承的 start 或者 end 值是根据父元素的 direction 值进行计算的，因此计算的结果可能是 left 或者 right
<string>	string 是一个单个的字符；否则，就忽略此设置。按指定的字符进行对齐。此属性可以跟其他关键字同时使用，如果没有设置字符，则默认值是 end 方式
inherit	继承父元素的对齐方式

在新增加的属性值中，start 和 end 属性值主要是针对行内元素的，即在包含元素的头部或尾部显示；而<string>属性值主要用于表格单元格中，将根据某个指定的字符对齐。

【例 5.12】设计段落中文本的水平对齐方式(案例文件：ch05\5.12.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>水平对齐方式</title>
</head>
<body>
<body>
<h1 style="text-align:center">关山月</h1>
<h3 style="text-align:left">选自：唐诗三百首</h3>
<h3 style="text-align:right">作者：李白</h3>
<p style="text-align:justify">
明月出天山，苍茫云海间。<br>
长风几万里，吹度玉门关。<br>
汉下白登道，胡窥青海湾。<br>
由来征战地，不见有人还。
</p>
<p style="text-align:start">戍客望边色，思归多苦颜。</p>
<p style="text-align:end">高楼当此夜，叹息未应闲。</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-12 所示。可以看到，文字在水平方向上以不同

的对齐方式显示。

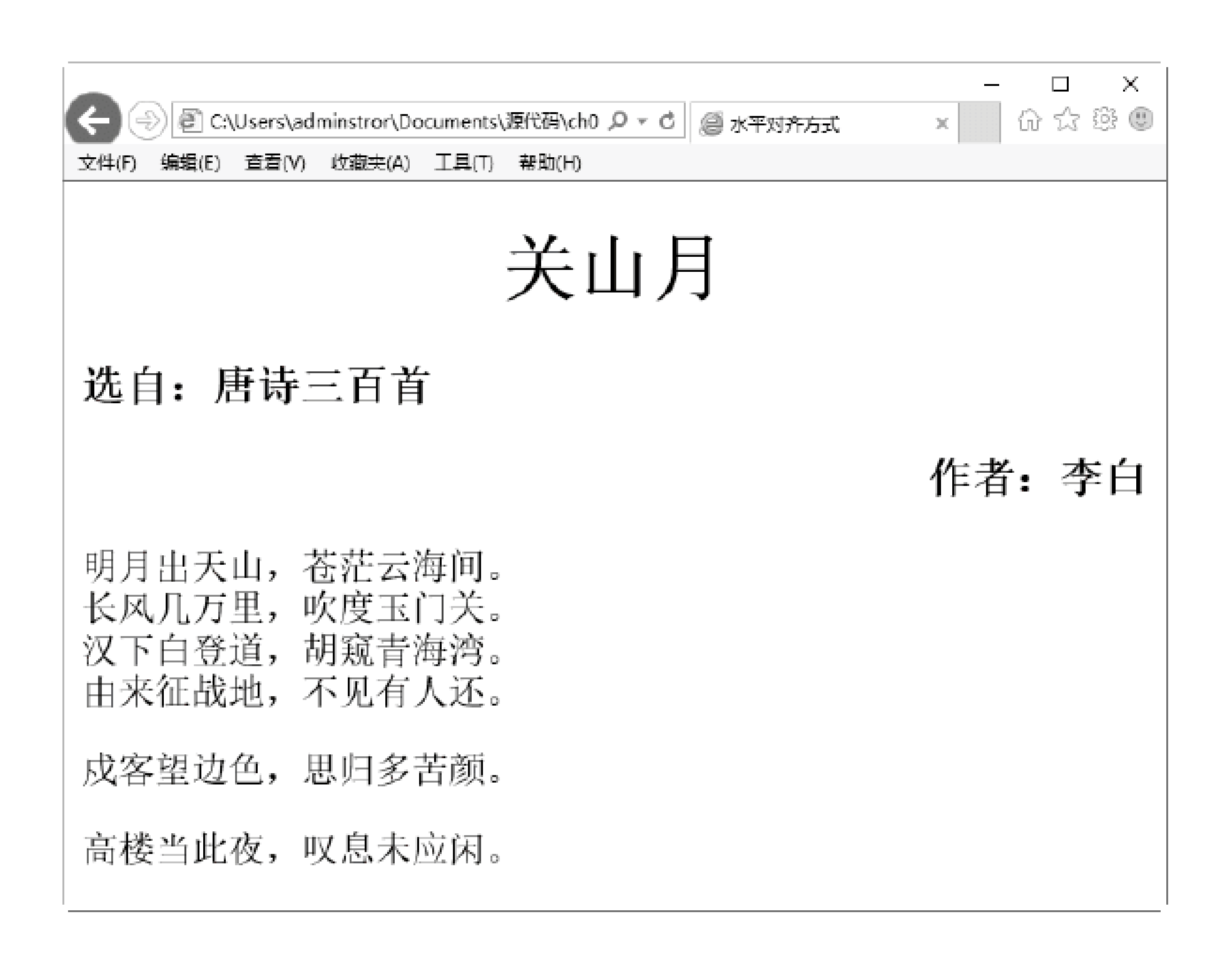


图 5-12　文本的水平对齐方式

text-align 属性只能用于文本块，而不能直接应用于图像标记<img>。如果要使图像同文本一样应用对齐方式，那么就必须将图像包含在文本块中。如上例，由于向右对齐方式作用于<h3>标记定义的文本块，图像包含在文本块中，所以图像能够同文本一样向右对齐。



默认只能定义两端对齐方式，并按要求显示。但对于具体的两端对齐文本如何分配字体空间以实现文本左右两边均对齐，并不规定。这就需要设计者自行定义了。

5.2.7　案例 13——设计文本缩进 text-indent

在普通段落中，通常首行缩进两个字符，用来表示这是一个段落的开始。同样，在网页的文本编辑中可以通过指定属性来控制文本缩进。使用 text-indent 属性可以设定文本块中首行的缩进。

text-indent 属性的语法格式如下：

```
text-indent : length
```

其中，length 属性值表示由百分比数字或由浮点数字和单位标识符组成的长度值，允许为负值。可以这样认为，text-indent 属性可以定义两种缩进方式：一是直接定义缩进的长度；二是定义缩进百分比。使用该属性，HTML 任何标记都可以让首行以给定的长度或百分比缩进。

【例 5.13】设计段落中文本的缩进方式(案例文件：ch05\5.13.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文本缩进</title>
</head>
```



```
<body>
这是一行默认文本。
<p style="text-indent:10mm">指定首行缩进 10mm。 </p>
<p style="text-indent:10%">首行缩进到当前行的百分之十。</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-13 所示。可以看到，文本以不同的首行缩进方式显示。

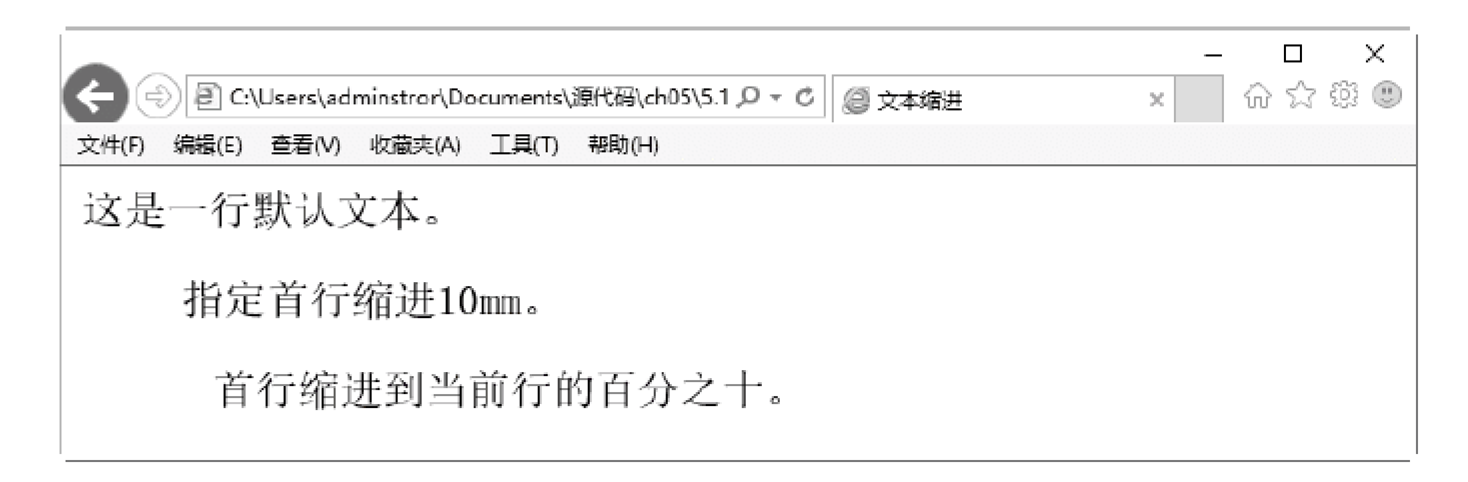


图 5-13 文本的缩进方式

如果上级标记定义了 text-indent 属性，那么子标记可以继承其上级标记的缩进长度。

5.2.8 案例 14——设计文本行高 line-height

在 CSS 中，line-height 属性用来设置行间距，即行高。其语法格式如下：

```
line-height: normal | length
```

line-height 属性值的具体含义如表 5-7 所示。

表 5-7 line-height 的属性值

属 性 值	说 明
normal	默认行高，即网页文本的标准行高
length	由百分比数字或由浮点数字和单位标识符组成的长度值，允许为负值。其百分比取值是基于字体的高度尺寸

【例 5.14】设计段落中文本的行高(案例文件：ch05\5.14.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文本行高</title>
</head>
<body>
<p style="line-height:50px">
文字列表可以有序地编排一些信息资源，使其结构化和条理化，并以列表的样式显示出来，以便浏览者能更加快捷地获得相应信息。HTML 中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。
</p>
<p style="line-height:70%">
文字列表可以有序地编排一些信息资源，使其结构化和条理化，并以列表的样式显示出来，以便浏览者能更加快捷地获得相应信息。HTML 中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。
</p>
```

```
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-14 所示。可以看到，有一段文字重叠在一起，即行高设置较小。

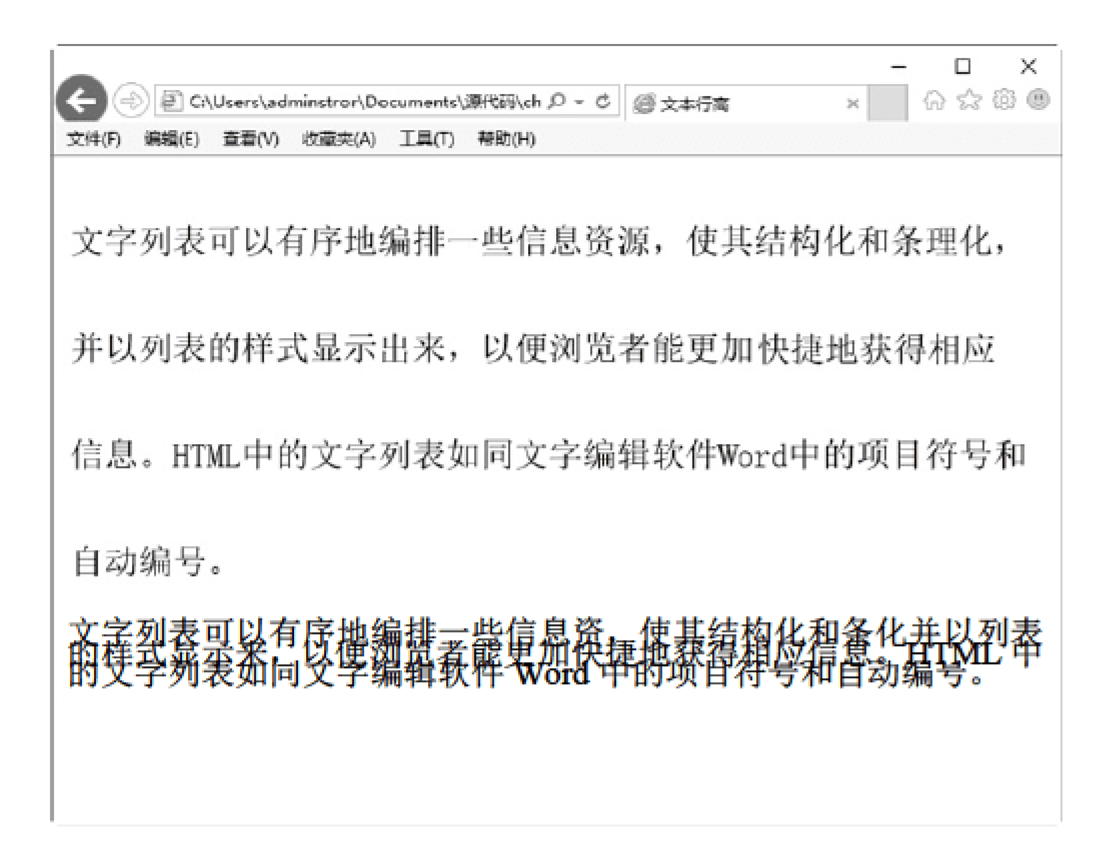


图 5-14　设定文本行高

5.2.9　案例 15——处理空白 white-space

在文本编辑中，网页中有时需要包含一些不必要的制表符、换行符或者额外的空白符(多于单词之间的一个标准的空格)，这些符号统称为空白字符。通常情况下，希望忽略这些额外的空白字符。浏览器可以自动完成此操作并按照一种适合窗口的方式布置文本。它会丢弃段落开头和结尾处任何额外的空白，并将单词之间的所有制表符、换行和额外的空白压缩(合并)成单一的空白字符。此外，当用户调整窗口大小时浏览器会根据需要重新格式化文本以便匹配新的窗口尺寸。对于某些元素，可能会以某种方式特意格式化文本以便包含额外的空白字符，而不希望抛弃或压缩这些字符。

使用 white-space 属性可以设置对象内空白字符的处理方式。white-space 属性对文本的显示有着重要的影响。在标记上应用 white-space 属性可以影响浏览器对字符串或文本间空白的处理方式。

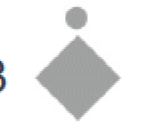
white-space 属性的语法格式如下：

```
white-space :normal | pre | nowrap | pre-wrap | pre-line
```

white-space 的属性值含义如表 5-8 所示。

表 5-8　white-space 的属性值

属 性 值	说　明
normal	默认。空白会被浏览器忽略
pre	空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签
nowrap	文本不会换行，文本会在同一行上继续，直到遇到 标签为止
pre-wrap	保留空白符序列，但是正常地进行换行



续表

属 性 值	说 明
pre-line	合并空白符序列，但是保留换行符
inherit	规定应该从父元素继承 white-space 属性的值

【例 5.15】设计段落中文本的留白样式(案例文件：ch05\5.15.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>处理空白</title>
</head>
<body>
    <h1 style="color:red; text-align:center;white-space:pre">网 页 文 字 列 表
设 计</h1>
    <p style="white-space:nowrap;text-indent:10mm">
    文字列表可以有序地编排一些信息资源，使其结构化和条理化，并以列表的样式显示出来，以便浏
览者能更加快捷地获得相应信息。<br>
    HTML 中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。
    </p>
    <p style="white-space:pre-wrap;text-indent:10mm">
    文字列表可以有序地编排一些信息资源，使其结构化和条理化，
            并以列表的样式显示出来，以便浏览者能更加快捷地获得相应
信息。<br/>
    HTML 中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。
    </p>
    <p style="white-space:pre-line;text-indent:10mm">
    文字列表可以有序地编排一些信息资源，使其结构化和条理化，并以列表的样式显示出来，以便浏
览者能更加快捷地获得相应信息。<br>
    HTML 中的文字列表如同文字编辑软件 Word 中的项目符号和自动编号。
    </p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-15 所示。可以看到，文字处理空白的不同方式。

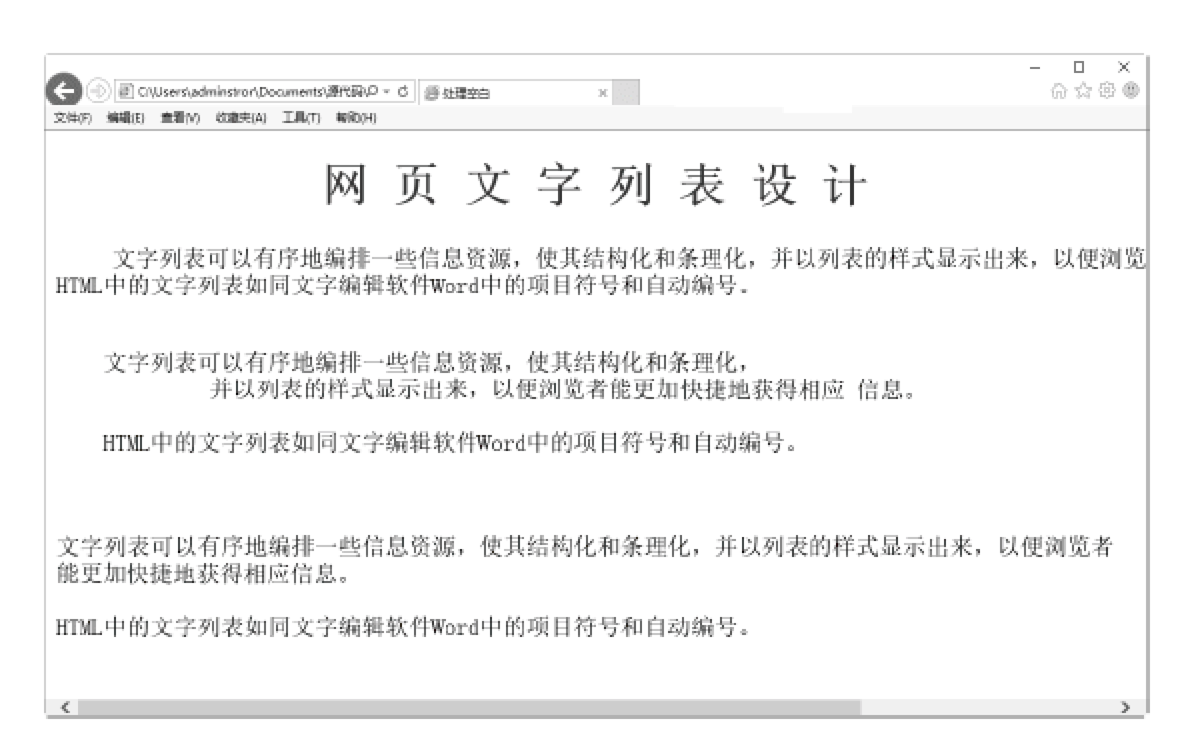
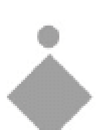


图 5-15　设定文本的留白效果

5.2.10　案例 16——文本反排 unicode-bidi

在网页文本编辑中，通常英语文档的基本方向是从左至右。如果文档中某一段的多个部



分包含从右至左阅读的语言，则该语言的方向将正确地显示为从右至左。使用 unicode-bidi 和 direction 两个属性可以解决文本反排的问题。

unicode-bidi 属性的语法格式如下：

```
unicode-bidi : normal | bidi-override | embed
```

unicode-bidi 属性值的含义如表 5-9 所示。

表 5-9　unicode-bidi 的属性值

属 性 值	说　明
normal	默认值。元素不会打开一个额外的嵌入级别。对于内联元素，隐式的重新排序将跨元素边界起作用
bidi-override	与 embed 值相同，但除了这一点外：在元素内，重新排序依照 direction 属性严格按顺序进行。此值替代隐式双向算法
embed	元素将打开一个额外的嵌入级别。direction 属性的值指定嵌入级别。重新排序在元素内是隐式进行的

direction 属性用于设定文本流的方向，其语法格式如下：

```
direction : ltr | rtl | inherit
```

direction 属性值的含义如表 5-10 所示。

表 5-10　direction 的属性值

属 性 值	说　明
ltr	文本流从左到右
rtl	文本流从右到左
inherit	文本流的值不可继承

【例 5.16】设计段落中文本的反排样式(案例文件：ch05\5.16.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文本反排</title>
</head>
<body>
<h2>文本反向排序显示</h2>
<p style=" direction:rtl; unicode-bidi:bidi-override; text-align:left">静坐常思己过，闲谈莫论人非。
</p>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 5-16 所示。可以看到，文字以反向显示。



图 5-16　设定文本的反排

5.3　综合案例——制作图文混排型旅游网页

在一个网页新闻中，出现最多的就是文字和图片，二者放在一起，图文并茂，能够生动地表达新闻主题。本实例将会利用前面介绍的文本和段落属性，创建一个图片的简单混排。复杂的图片混排，会在后面介绍。具体操作步骤如下。

1. 分析需求

本综合实例的要求如下：在网页的最上方显示出标题，标题下方是正文，在正文显示部分显示图片。其实例效果图如图5-17所示。



图5-17　图文混排旅游网页

2. 实现代码

step 01 打开记事本，编写HTML代码基本框架。具体代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>图文混排网页</title>
</head>
<body>
</body>
</html>
```

step 02 在<body>标签中插入网页标题设计代码：

```
<h1 style="text-align:center;text-shadow:0.1em 2px 6px blue;font-
size:18px">塞外江南：伊犁哈萨克自治州</h1>
```

step 03 在<body>标签中插入图片设计代码：

```
<img src="images/3.jpg" style="text-align:center;width:300px; float:left;
border:#000000 solid 2px">
```

step 04 在<body>标签中完善文本段落内容设计代码：

```
<p style="text-indent:8mm;line-height:7mm">伊犁哈萨克自治州地处祖国西北边陲，成立于1954年，辖塔城、阿勒泰两个地区和10个直属县市，是全国唯一的既辖地区又辖县市的自治州。西部紧邻欧亚国家哈萨克斯坦，这里有中国陆路最大的通商口岸(霍尔果斯口岸)。
</p>
<p style="text-indent:8mm;line-height:7mm">伊犁哈萨克自治州，地处祖国西北边陲，气候宜人，是中国十大宜居中小城市之一，降水量较为丰富。</p>
<img src="images/3.jpg" style="text-align:center;width:300px; float:left;
border:#000000 solid 2px">
<p style="text-indent:8mm;line-height:7mm">自治州境内驻有普通高等院校伊犁师范学院、新疆生产建设兵团农业第四、七、八、九、十师和新疆矿冶局、天西林业局、阿山林业局、新疆卷烟厂、阿希金矿等一批中央和自治区直属单位。伊犁被誉为“塞外江南”、“中亚湿岛”，“花城”伊宁市是伊犁州的首府。<br>
伊犁幅员辽阔，资源充裕，有着得天独厚的优势。伊犁的矿产资源种类齐全。生物资源十分珍贵。森林面积88万公顷，活立木总蓄积量1.6亿立方米，占全疆的74%；保存着60多种珍稀动物，700多种植物，是世界上少有的生物多样性天然基因库，具有很高的科学研究和开发利用价值。旅游资源独特。地理、水体、生物景观和文物古迹、民俗风情、休闲健身等六大旅游资源类型一应俱全。有美丽的草原风光，浓郁的民俗风情，独特的草原文化，悠久的历史古迹，是中国西部最理想的旅游目的地。
</p>
```

step 05 保存编辑好的网页文件，保存名称为“图文混排网页.html”。

5.4 高手解惑

疑问1：无序列表<ul>元素的作用？

答：无序列表元素主要用于条理化和结构化文本信息。在实际开发中，无序列表在制作导航菜单时使用广泛。导航菜单的结构一般都使用无序列表实现。

疑问2：文字和图片导航速度谁更快？

答：使用文字做导航栏。文字导航不仅速度快，而且更稳定。例如，有些用户上网时会关闭图片。在处理文本时，不要在普通文本上添加下画线或者颜色。除非特别需要，否则不要为普通文字添加下画线。就像用户需要识别哪些能点击一样，读者不应当将本不能点击的文字误认为能够点击。

第 6 章 HTML 5 网页中的图像

图像是网页中最主要也是最常用的元素。图像在网页中具有画龙点睛的作用。它能装饰网页，表达个人的情调和风格。但在网页上加入的图片越多，浏览的速度就越会受到影响，可能导致用户失去耐心而离开页面。本章介绍如何使用 HTML 5 为网页插入图像。

重点案例效果



6.1 网页中的图像<img>

俗话说“一图胜千言”，图片是网页中不可或缺的元素，巧妙地在网页中使用图片可以为网页增色不少。网页支持多种图片格式，并且可以对插入的图片设置宽度和高度。

6.1.1 网页中支持的图片格式

网页中可以使用 GIF、JPEG、BMP、TIFF、PNG 等格式的图像文件，其中使用最广泛的主要是 GIF 和 JPEG 两种格式。

1. GIF 格式

GIF 格式是由 Compuserve 公司提出的与设备无关的图像存储标准，也是 Web 上使用最早、应用最广泛的图像格式。GIF 是通过减少组成图像的每个像素的储存位数和 LZH 压缩存储技术来减少图像文件的大小。GIF 格式最多只能是 256 色的图像。

GIF 图像文件短小，下载速度快，低颜色数下 GIF 比 JPEG 载入得更快，可用许多具有同样大小的图像文件组成动画。在 GIF 图像中可指定透明区域，使图像具有非同一般的显示效果。

2. JPEG 格式

JPEG 格式是目前 Internet 中最受欢迎的图像格式。它可以支持多达 16M 的颜色，能展现十分丰富生动的图像，还能压缩。但其压缩方式是以损失图像质量为代价的，压缩比越高，图像质量损失越大，图像文件也就越小。

位图 BMP 格式的图像是 Windows 支持的较为流行的图像格式。一般情况下，同一图像的 BMP 格式的大小是 JPEG 格式的 5～10 倍。而 GIF 格式最多只能是 256 色，因此载入 256 色以上图像的 JPEG 格式成了 Internet 中最受欢迎的图像格式。

当网页中需要载入一个较大的 GIF 或 JPEG 图像文件时，载入速度会很慢。为改善网页的视觉效果，可以在载入时设置为隔行扫描。隔行扫描在显示图像开始时看起来非常模糊，接着细节逐渐添加上去，直到图像完全显示出来。

GIF 是支持透明、动画的图片格式，但色彩只有 256 色。JPEG 是一种不支持透明和动画的图片格式，但是色彩模式比较丰富，保留大约 1670 万种颜色。



网页中现在也有很多 PNG 格式的图片。PNG 图片具有不失真、兼有 GIF 和 JPG 的色彩模式、网络传输速度快、支持透明图像制作的特点，近年来在网络中也很流行。

6.1.2 图像中的路径

HTML 文档支持文字、图片、声音、视频等媒体格式。但是在这些格式中，除了文本是

写在 HTML 中的，其他都是嵌入式的，HTML 文档只记录了这些文件的路径。这些媒体信息能否正确显示，路径至关重要。

路径的作用是定位一个文件的位置。文件的路径可以有两种表述方法，以当前文档为参照物表示文件的位置，即相对路径。以根目录为参照物表示文件的位置，即绝对路径。

为了方便讲述绝对路径和相对路径，先看如图 6-1 所示的目录结构。

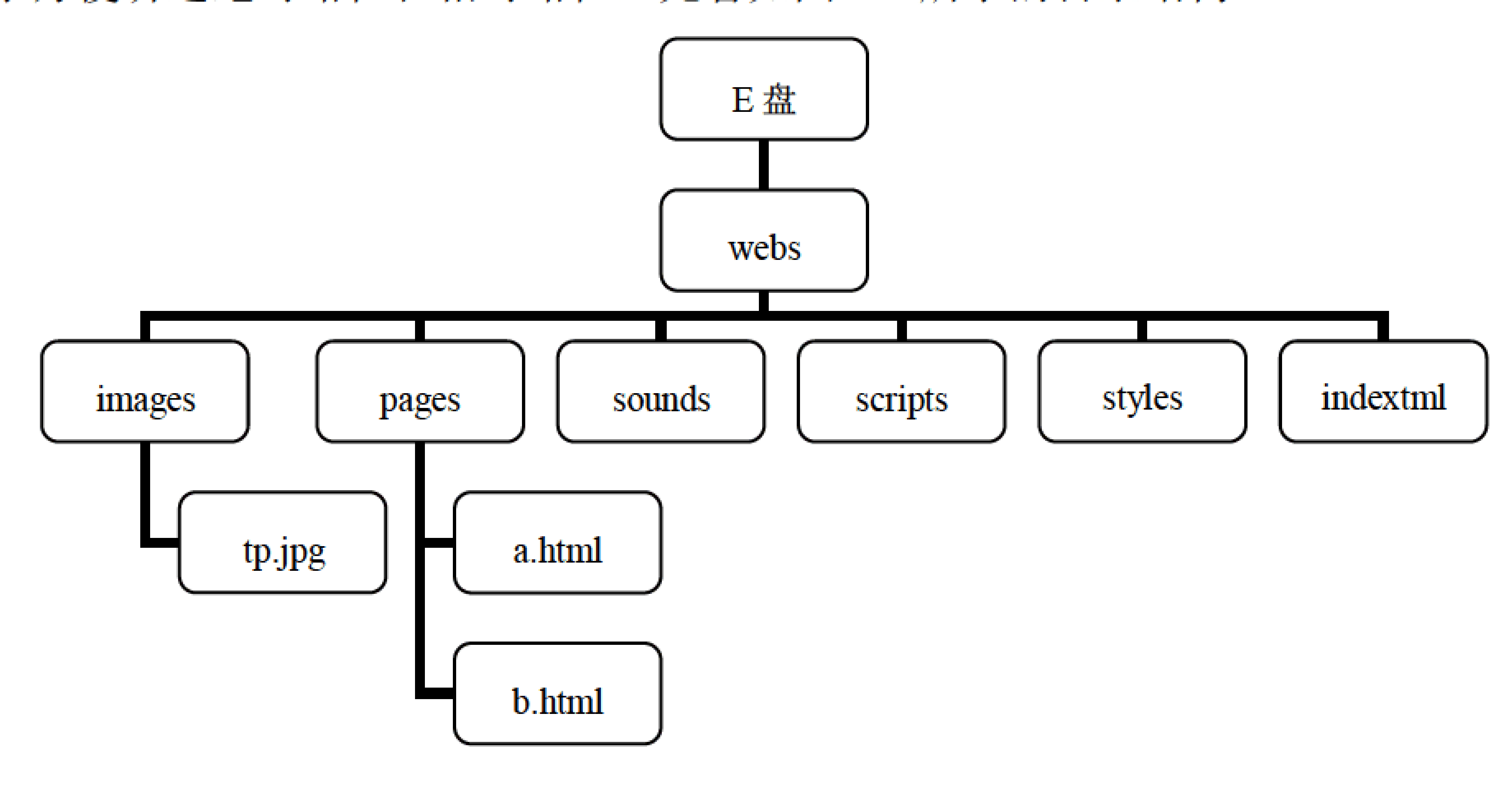


图 6-1 目录结构

1. 绝对路径

例如，在 E 盘的 webs 目录下的 images 下有一个 tp.jpg 图像，那么它的路径就是 E:\webs\images\tp.jpg，像这种完整地描述文件位置的路径就是绝对路径。如果将图片文件 tp.jpg 插入到网页 index.html 中，绝对路径表示方式如下：

```
E:\webs\images\tp.jpg
```

如果使用了绝对路径 E:\webs\images\tp.jpg 进行图片链接，那么在本地电脑中将一切正常，因为在 E:\webs\images 下的确存在 tp.jpg 这个图片。如果将文档上传到网站服务器上后，那就会不正常了，因为服务器给你划分的存放空间可能在 E 盘其他目录中，也可能在 D 盘其他目录中。为了保证图片正常显示，必须从 webs 文件夹开始，放到服务器或其他电脑的 E 盘根目录下。

通过上述讲解，读者会发现，如果链接的资源是本站点内的使用绝对路径，对位置要求非常严格。因此，链接本站点内的资源不建议采用绝对路径。如果链接其他站点的资源，必须使用绝对路径。

2. 相对路径

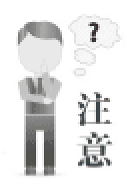
如何使用相对路径设置上述图片呢？所谓相对路径，顾名思义就是以当前位置为参考点，自己相对于目标的位置。例如，在 index.html 中链接 tp.jpg 就可以使用相对路径。index.html 和 tp.jpg 图片的路径根据上述目录结构图可以这样来定位：从 index.html 位置出发，它和 images 属于同级，路径是通的，因此可以定位到 images，images 的下级就是 tp.jpg。使用相对路径表示图片如下：

```
images/tp.jpg
```

使用相对路径，不论将这些文件放到哪里，只要 tp.jpg 和 index.html 文件的相对关系没有变，就不会出错。

在相对路径中，“..”表示上一级目录，“../..”表示上级的上级目录，以此类推。例如，将 tp.jpg 图片插入到 a.html 文件中，使用相对路径表示如下：

```
../images/tp.jpg
```



细心的读者会发现，路径分隔符使用了“\”和“/”两种，其中“\”表示本地分隔符，“/”表示网络分隔符。因为网站制作好后肯定是在网络上运行的，所以要求使用“/”作为路径分隔符。

有的读者可能会有这样的疑惑：一个网站有许多链接，怎么能保证它们的链接都正确，如果修改了图片或网页的存储路径，那不是全乱了吗？如何提高工作效率呢？



Dreamweaver 工具的站点管理功能，不但可以将绝对路径自动地转化为相对路径，并且当在站点中改动文件路径时，与这些文件关联的链接路径都会自动更改。

6.2 在网页中插入图像

图像可以美化网页。插入图像使用单标记<img>。img 标记的属性及其描述如表 6-1 所示。

表 6-1 img 标记的属性及其描述

属　性	值	描　述
alt	text	定义有关图形的简短的描述
src	URL	要显示的图像的 URL
height	pixels %	定义图像的高度
ismap	URL	把图像定义为服务器端的图像映射
usemap	URL	定义作为客户端图像映射的一幅图像。请参阅 <map> 和 <area> 标签，了解其工作原理
vspace	pixels	定义图像顶部和底部的空白。不支持。请使用 CSS 代替
width	pixels %	设置图像的宽度

6.2.1 案例 1——插入图像

src 属性用于指定图像源文件的路径，它是 img 标记必不可少的属性。其语法格式如下：

```
<img src="图像路径">
```

图像的路径可以是绝对路径，也可以是相对路径。下面的实例是在网页中插入图像。

【例 6.1】 在网页中插入图像(案例文件：ch06\6.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>插入图片</title>
</head>
<body>
<img src="images/01.jpg">
</body>
</html>
```

在 IE 11.0 中预览，效果如图 6-2 所示。

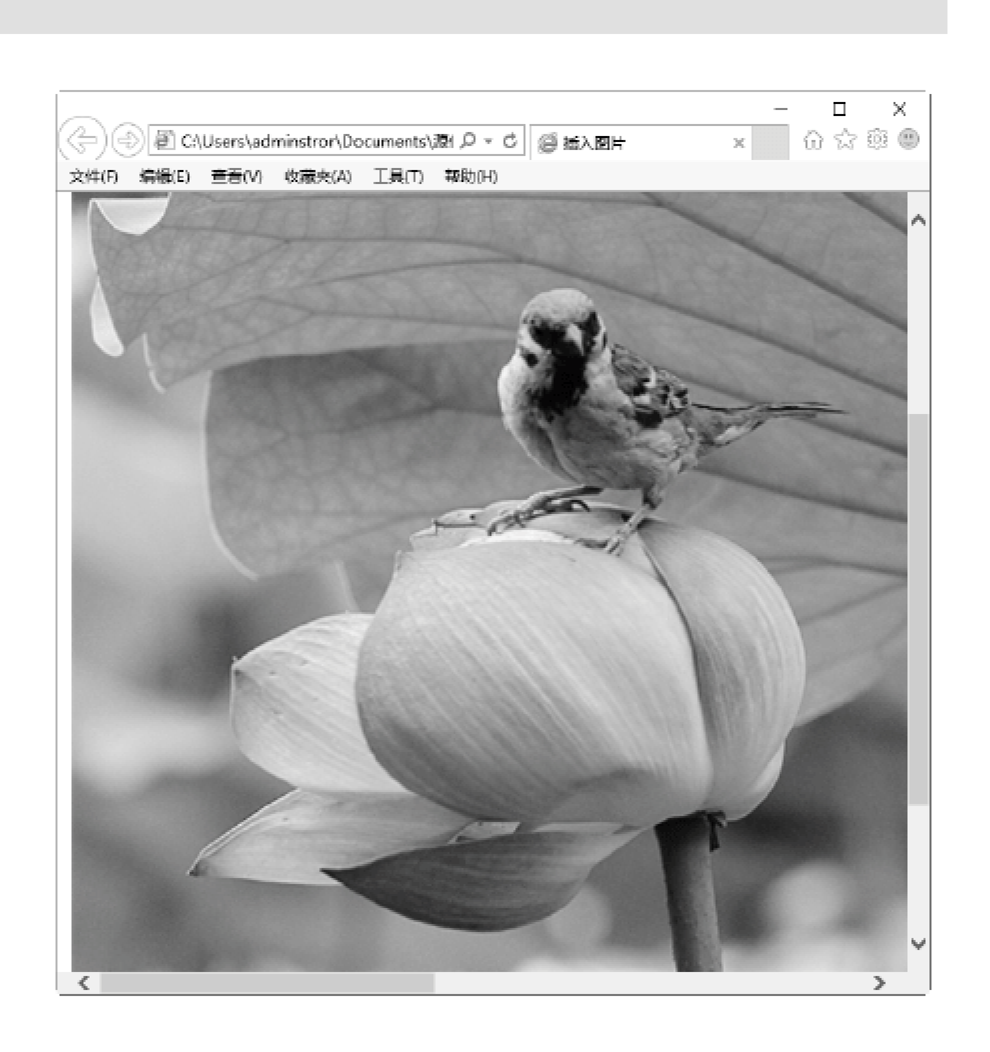


图 6-2　插入图像

6.2.2　案例 2——从不同位置插入图像

在插入图片时，用户可以将其他文件夹或服务器的图片显示到网页中。

【例 6.2】 从不同位置插入图像(案例文件：ch06\6.2.html)。

```
<!DOCTYPE html>
<html>
<body>
<p>
来自一个文件夹的图像：
<img src="images/meishi.jpg" />
</p>
<p>
来自 baidu 的图像：
<img
src="http://www.baidu.com/img/shouye b548
6898c692066bd2cbaeda86d74448.gif" />
</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 6-3 所示。



图 6-3　从不同位置插入图像

6.3　编辑网页中的图像

在插入图片时，用户还可以编辑网页中的图像。

6.3.1　案例 3——设置图像的宽度和高度

在 HTML 文档中，还可以设置插入图片的显示大小，一般是按原始尺寸显示，但也可以任意设置显示尺寸。设置图像尺寸分别用属性 width(宽度)和 height(高度)。

【例 6.3】设置图像的宽度和高度(案例文件：ch06\6.3.html)。

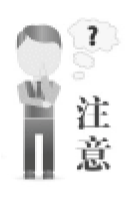
```
<!DOCTYPE html>
<html>
<head>
<title>设置图像的宽度和高度</title>
</head>
<body>
<img src="images/meishi.jpg">
<img src="images/meishi.jpg" width="200">
<img src="images/meishi.jpg" width="200" height="300">
</body>
</html
```

在 IE 11.0 中预览，效果如图 6-4 所示。

由图 6-4 可以看到，图片的显示尺寸是由 width(宽度)和 height(高度)控制。当只为图片设置一个尺寸属性时，另一个尺寸就以图片原始的长宽比例来显示。图片的尺寸单位可以选择百分比或数值。百分比是相对尺寸，数值是绝对尺寸。



图 6-4　设置图像的宽度和高度



对于网页中插入的图像都是位图，放大尺寸，图像会出现马赛克，变得模糊。



在 Windows 中查看图片的尺寸，只需要找到图像文件，把鼠标指针移动到图像上，停留几秒后，就会出现一个提示框，说明图像文件的尺寸。尺寸后显示的数字，代表图像的宽度和高度，如 256×256。

6.3.2　案例 4——设置图像的提示文字

图像提示文字的作用有两个。其一，当浏览网页时，如果图像下载完成，将鼠标指针放

在该图像上，鼠标指针旁边会出现提示文字，为图像添加说明性文字。其二，如果图像没有成功下载，在图像的位置上就会显示提示文字。

为图像添加提示文字可以方便搜索引擎的检索。除此之外，图像提示文字的作用还有以下两个。

(1) 当浏览网页时，如果图像下载完成，将鼠标指针放在该图像上，鼠标指针旁边会显示 title 标签设置的提示文字。

(2) 如果图像没有成功下载，在图像的位置上会显示 alt 标记设置的提示文字。

下面的实例将为图片添加提示文字效果。

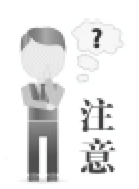
【例 6.4】设置图像的提示文字(案例文件：ch06\6.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>图片文字提示</title>
</head>
<body>
<img src="images/meishi.jpg" alt="未加载完成时显示的替代文字"  title="草莓甜橙沙拉">
</body>
</html>
```

在 IE 11.0 中预览，效果如图 6-5 所示。用户将鼠标放在图片上，即可看到提示文字。



图 6-5　图片文字提示



随着互联网技术的发展，网速已经不是制约因素，因此一般都能成功下载图像。现在，alt 还有另外一个作用，即在百度、Google 等搜索引擎中，搜索图片不如搜索文字方便，如果给图片添加适当提示，可以方便搜索引擎的检索。

6.3.3　案例 5——将图片设置为网页背景

在插入图片时，用户可以根据需要将某些图片设置为网页的背景。GIF 和 JPG 文件均可用作 HTML 背景。如果图像小于页面，图像会进行重复。

【例 6.5】将图片设置为网页背景(案例文件：ch06\6.5.html)。

```
<!DOCTYPE html>
<html>
<body background="images/background.jpg">
<h3>图像背景</h3>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 6-6 所示。



图 6-6　图像背景

6.3.4　案例 6——排列图像

在网页的文字中，如果插入图片，这时可以对图像进行排序。常用的排序方式有居中、底部对齐、顶部对齐三种。

【例 6.6】排列图像(案例文件：ch06\6.6.html)。

```
<!DOCTYPE html>
<html>
<body>
<h2>未设置对齐方式的图像：</h2>
<p>图像<img src ="images/logo.gif"> 在文本中</p>
<h2>已设置对齐方式的图像：</h2>
<p>图像 <img src=" images/logo.gif " align="bottom"> 在文本中</p>
<p>图像 <img src =" images/logo.gif " align="middle"> 在文本中</p>
<p>图像 <img src =" images/logo.gif " align="top"> 在文本中</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 6-7 所示。



图 6-7　图片对齐方式



bottom 对齐方式是默认的对齐方式。

6.4 综合案例——图文并茂的房屋装饰装修网页

本章讲述了网页组成元素中最常用的文本和图片。本综合实例将创建一个由文本和图片构成的房屋装饰效果网页，如图 6-8 所示。



图 6-8 房屋装饰效果网页

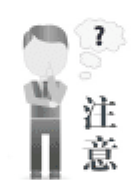
具体操作步骤如下。

step 01 在 Dreamweaver CC 中新建 HTML 5 文档。代码如下：

```
<!DOCTYPE html>
<html >
<head>
<title>房屋装饰装修效果图</title>
</head>
<body>
</body>
</html>
```

step 02 在 body 部分增加如下 HTML 代码，保存页面。

```
<p> <img src="images/xiyatu.jpg" width="300" height="200"/> <img
src="images/stadshem.jpg" width="300" height="200"/><br />
西雅图原生态公寓室内设计 与 Stadshem 小户型公寓设计(带阁楼)</p>
<hr/>
<p> <img src="images/qingxinhuoli.jpg" width="300" height="200"/> <img
src="images/renwen.jpg" width="300" height="200"/><br />
清新活力家居与人文简约悠然家居</p>
<hr />
```



<hr>标记的作用是定义内容中的主题变化，并显示为一条水平线，在 HTML 5 中它没有任何属性。

另外，快速插入图片及设置相关属性，可以借助 Dreamweaver CC 的插入功能，或按 Ctrl+Alt+I 组合键。

6.5 高手解惑

疑问 1：在浏览器中，图片无法正常显示，为什么？

答：图片在网页中属于嵌入对象，并不是图片保存在网页中，网页只是保存了指向图片的路径。浏览器在解释 HTML 文件时，会按指定的路径去寻找图片，如果在指定的位置不存在图片，就无法正常显示。为了保证图片的正常显示，制作网页时需要注意以下几点。

(1) 图片格式一定是网页支持的。

(2) 图片的路径一定要正常，并且图片文件扩展名不能省略。

(3) HTML 文件位置发生改变时，图片一定要跟随着改变，即图片位置和 HTML 文件位置始终保持相对一致。

疑问 2：在网页中，有时使用图像的绝对路径，有时使用相对路径，为什么？

答：如果在同一个文件中需要反复使用一个相同的图像文件时，最好在<img>标签中使用相对路径名，不要使用绝对路径名或 URL，因为，使用相对路径名，浏览器只需将图像文件下载一次，再次使用这个图像时，只要重新显示一遍即可。如果使用绝对路径名，每次显示图像时，都要下载一次图像，这将大大降低图像的显示速度。



第 7 章 使用 HTML 5 建立超链接

HTML 文件中最重要的应用之一就是超链接。超链接是一个网站的灵魂。Web 上的网页是互相链接的，单击被称为超链接的文本或图形就可以链接到其他页面。只有将网站中的各个页面链接在一起之后，这个网站才能被称为真正的网站。



7.1　网页超链接的概念

所谓超链接，是指从一个网页指向一个目标的链接关系。这个目标可以是另一个网页，也可以是相同网页上的不同位置，还可以是一个图片、一个电子邮件地址、一个文件，甚至是一个应用程序。

7.1.1　什么是网页超链接

超链接是一种对象，它以特殊编码的文本或图形的形式来实现链接。如果单击该链接，则相当于指示浏览器移至同一网页内的某个位置，或打开一个新的网页，或打开某一个新的WWW 网站中的网页。

网页中的链接按照链接路径的不同，可以分为 3 种类型，分别是内部链接、锚点链接和外部链接。按照使用对象的不同，网页中的链接又可以分为文本超链接、图像超链接、E-mail链接、锚点链接、多媒体文件链接、空链接等。

在网页中，一般文字上的超链接都是蓝色，文字下面有一条下画线。当移动鼠标指针到该超链接上时，鼠标指针就会变成一只手的形状，这时用鼠标左键单击，就可以直接跳转到与这个超链接相连接的网页或 WWW 网站上去。如果用户已经浏览过某个超链接，这个超链接的文本颜色就会发生改变(默认为紫色)。只有图像的超链接访问后颜色不会发生变化。

7.1.2　超链接中的 URL

URL 为 Uniform Resource Locator 的缩写，通常翻译为“统一资源定位器”，也就是人们通常说的“网址”，它用于指定 Internet 上的资源位置。

网络中的计算机之间是通过 IP 地址区分的。如果希望访问网络中某台计算机中的资源，首先要定位到这台计算机。IP 地址是由 32 位二进制数(即 32 个 0/1 代码)组成的，数字之间没有意义，不容易记忆。为了方便记忆，现在计算机一般采用域名的方式来寻址，即在网络上使用一组有意义字符组成的地址代替 IP 地址来访问网络资源。

URL 由 4 个部分组成，即“协议”“主机名”“文件夹名”“文件名”，如图 7-1 所示。



图 7-1　URL 组成

互联网中有各种各样的应用，如 Web 服务、FTP 服务等。每种服务应用都有对应的协议，通常通过浏览器浏览网页的协议都是 HTTP 协议，即“超文本传输协议”，因此网页的地址都以 http://开头。

www.baidu.com 为主机名，表示文件存在于哪台服务器，主机名可以通过 IP 地址或者域名来表示。

确定主机后，还需要说明文件存在于这台服务器的哪个文件夹中，这里文件夹可以分为多个层级。

确定文件夹后，就要定位到文件，即要显示哪个文件，网页文件通常是以.html 或.htm 为扩展名。

7.1.3 超链接的 URL 类型

网页上的超链接一般分为 3 种，分别介绍如下。

(1) 绝对 URL 超链接。URL 就是统一资源定位符，简单地讲就是网络上的一个站点、网页的完整路径。

(2) 相对 URL 超链接。如将自己网页上的某一段文字或某标题链接到同一网站的其他网页上面去。

(3) 书签超链接。同一网页的超链接，这种超链接又叫作书签。

7.2 建立网页超级链接

超级链接就是当鼠标单击一些文字、图片或其他网页元素时，浏览器就会根据其指示载入一个新的页面或跳转到页面的其他位置。超级链接除了可以链接文本外，也可以链接各种媒体，如声音、图像、动画，通过它们可享受丰富多彩的多媒体世界。

建立超级链接所使用的 HTML 标记为<a>。超级链接最重要的有两个要素，即设置为超级链接的网页元素和超级链接指向的目标地址。基本的超级链接的结构如下：

```
<a href=URL>网页元素</a>
```

7.2.1 案例 1——创建超文本链接

文本是网页制作中使用最频繁也是最主要的元素。为了实现跳转到与文本相关内容的页面，往往需要为文本添加链接。

1. 什么是文本链接

浏览网页时，会看到一些带下画线的文字，将鼠标指针移到文字上时，鼠标指针将变成手形，单击会打开一个网页，这样的链接就是文本链接，如图 7-2 所示。

2. 创建链接的方法

使用<a>标签可以实现网页超链接，在<a>标签处需要定义锚来指定链接目标。锚(anchor)有两种用法，分别介绍如下。

(1) 通过使用 href 属性，创建指向另外一个文档的链接(或超链接)。使用 href 属性的代码格式如下：

```
<a href="链接地址">创建链接的文本</a>
```



图 7-2　存在文本链接的网页

(2) 通过使用 name 或 id 属性，创建一个文档内部的书签(也就是说，可以创建指向文档片段的链接)。使用 name 属性的代码格式如下：

```
<a name="value">创建链接的文本</a>
```

name 属性用于指定锚的名称。name 属性可以创建(大型)文档内的书签。

使用 id 属性的代码格式如下：

```
<a id="value">创建链接的文本</a>
```

3．创建网页内的文本链接

创建网页内的文本链接主要使用 href 属性来实现。比如，在网页中做一些知名网站的友情链接。

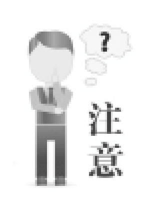
【例 7.1】使用记事本创建网页超文本链接(案例文件：ch07\7.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文本链接</title>
</head>
<body>
友情链接————
<a href="http://www.baidu.com">百度</a>
<a href="http://www.sina.com.cn">新浪</a>
<a href="http://www.163.com">网易</a>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 7-3 所示，带有超链接的文本呈现浅蓝色。



图 7-3　创建的文本链接网页效果



注意

链接地址前的“http://”不可省略，否则链接会出现错误提示。

7.2.2 案例 2——创建图片链接

在网页中浏览内容时，若将鼠标指针移动到图片上，鼠标指针将变成手形，单击会打开一个网页，这样的链接就是图片链接，如图 7-4 所示。



图 7-4　存在图片链接的网页

使用<a>标签为图片添加链接的代码格式如下：

```
<a href="链接目标"><img src="图片"/></a>
```

【例 7.2】使用记事本创建网页图片链接(案例文件：ch07\7.2.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>图片链接</title>
</head>
<body>
音乐无限
<a href="mp3.html"><img src="1.jpg"/></a>
<br>
<br>
<br>
运动健身
<a href="tiyu.html"><img src="2.jpg"/></a>
</body>
</html>
```

使用 IE 11.0 打开文件预览，效果如图 7-5 所示，鼠标指针放在图片上呈现手指状，单击后可以跳转到指定网页。



图 7-5　创建的图片链接网页效果



文件中的图片要和当前网页文件在同一目录下，链接的网页没有加“http://”，默认为当前网页所在目录。

7.2.3　案例 3——创建下载链接

超链接<a>标记 href 属性是指向链接的目标，目标可以是各种类型的文件，如图片文件、声音文件、视频文件、Word 等。如果是浏览器能够识别的类型，会直接在浏览器中显示。如果是浏览器不能识别的类型，在 IE 浏览器中会弹出文件下载对话框，如图 7-6 所示。



图 7-6　IE 中的文件下载对话框

【例 7.3】创建下载链接(案例文件：ch07\7.3.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>链接各种类型文件</title>
</head>
<body>
<p><a href="2.doc">链接 word 文档</a></p>
</body>
</html>
```



图 7-7　链接 Word 文档

在 IE 11.0 中预览，效果如图 7-7 所示。实现链



接到 html 文件、图片和 Word 文档。

7.2.4 案例 4——使用相对路径和绝对路径

绝对 URL 一般位于访问非同一台服务器上的资源；相对 URL 是指访问同一台服务器上相同文件夹或不同文件夹中的资源。如果访问相同文件夹中的文件，只需要写文件名；如果访问不同文件夹中的资源，URL 以服务器的根目录为起点，指明文档的相对关系，由文件夹名和文件名两个部分构成。

【例 7.4】使用绝对 URL 和相对 URL 实现超链接(案例文件：ch07\7.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绝对 URL 和相对 URL</title>
</head>
<body>
  单击<a href="http://www.webDesign.com/index.html">绝对 URL</a>链接到
webDesign 网站首页<br />
  单击<a href="02.html">相同文件夹的 URL</a>链接到相同文件夹中的第 2 个页面<br />
  单击<a href="../pages/03.html">不同文件夹的 URL</a>链接到不同文件夹中的第 3 个页面
</body>
</html>
```

在上述代码中，第 1 个链接使用的是绝对 URL；第 2 个使用的是服务器相对 URL，也就是链接到文档所在服务器的根目录下的 02.html 文件；第 3 个使用的是文档相对 URL，即原文档所在文件夹的父文件夹下面的 pages 文件夹中的 03.html 文件。

在 IE 11.0 中预览，效果如图 7-8 所示。

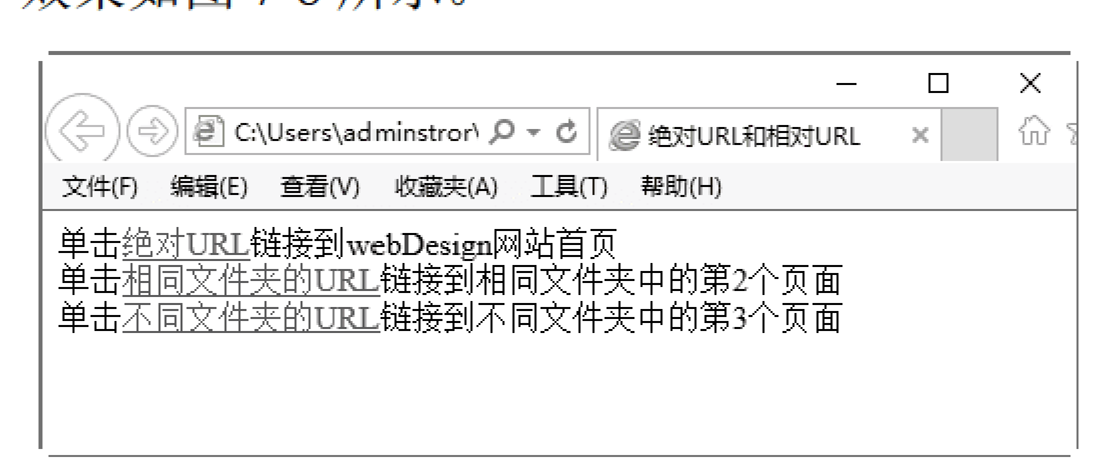


图 7-8 绝对 URL 和相对 URL

7.2.5 案例 5——设置以新窗口显示超链接页面

在默认情况下，当单击超链接时，目标页面会在当前窗口中显示，替换当前页面的内容。如果要在单击某个链接以后，打开一个新的浏览器窗口在这个新窗口中显示目标页面，就需要使用<a>标签的 target 属性。

target 属性的代码格式如下：

```
<a target="value">
```

其中，value 有 4 个参数可用，这 4 个保留的目标名称用作特殊的文档重定向操作。

(1) _blank。浏览器总在一个新打开、未命名的窗口中载入目标文档。

(2) _self。这个目标的值对所有没有指定目标的 <a> 标签是默认目标，它使得目标文档载入并显示在相同的框架或者窗口中作为源文档。这个目标是多余且不必要的，除非和文档标题<base> 标签中的 target 属性一起使用。

(3) _parent。这个目标使得文档载入父窗口或者包含在超链接引用的框架的框架集。如果这个引用是在窗口或者顶级框架中，那么它与目标 _self 等效。

(4) _top。这个目标使得文档载入包含这个超链接的窗口，用_top 目标将会清除所有被包含的框架并将文档载入整个浏览器窗口。

【例 7.5】设置以新窗口显示超链接页面(案例文件：ch07\7.5.html)。



```
<!DOCTYPE html>
<html>
<head>
<title>设置链接目标</title>
</head>
<body>
<a href="http://www.baidu.com"
target="_blank">百度</a>
</body>
</html>
```

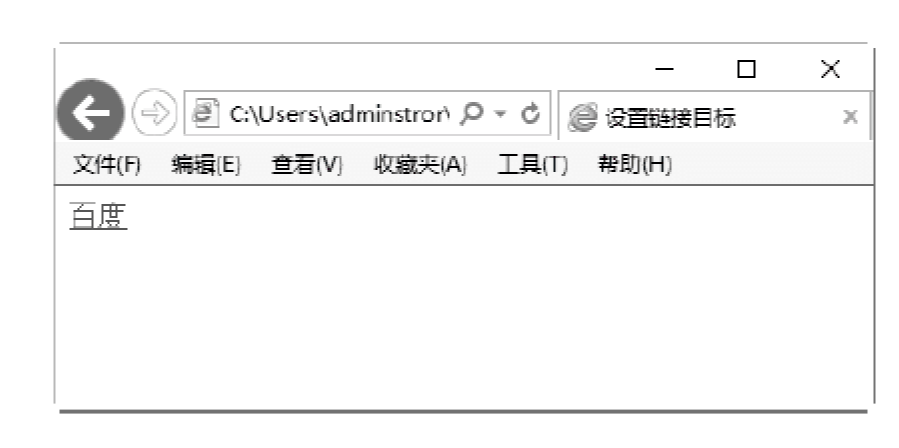


图 7-9 制作网页超链接

使用 IE 11.0 打开网页文件，显示效果如图 7-9 所示。

单击网页中的超链接，在新窗口中打开链接页面，如图 7-10 所示。

如果将_blank 换成_self，即代码修改为“百度”，单击链接后，则直接在当前窗口中打开新链接，如图 7-11 所示。





图 7-10 在新窗口中打开链接网页



图 7-11 在当前窗口中打开链接网页



target 的 4 个值都以下画线开始。任何其他用一个下画线作为开头的窗口或者目标都会被浏览器忽略。因此，不要将下画线作为文档中定义的任何框架 name 或 id 的第一个字符。

7.2.6 案例 6——设置电子邮件链接

在某些网页中，当访问者单击某个链接以后，会自动打开电子邮件客户端软件，如 Outlook 或 Foxmail 等，向某个特定的 E-mail 地址发送邮件，这个链接就是电子邮件链接。电

子邮件链接的格式如下：

```
<a href="mailto:电子邮件地址" >网页元素</a>
```

【例 7.6】设置电子邮件链接(案例文件：ch07\7.6.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>电子邮件链接</title>
</head>
<body>
<img src="images/logo.gif" width="119" height="49">  [免费注册][登录]
<a href="mailto:kfdzsj@126.com">站长信箱</a>
</body>
</html>
```

在 IE 11.0 中预览网页，效果如图 7-12 所示，实现了电子邮件链接。



图 7-12 链接到电子邮件

当读者单击“站长信箱”链接时，会自动弹出 Outlook 窗口，要求编写电子邮件，如图 7-13 所示。



图 7-13 Outlook 新邮件窗口

7.3 案例 7——浮动框架 iframe 的使用

HTML 5 中已经不支持 frameset 框架，但是它仍然支持 iframe 浮动框架的使用。浮动框

架可以自由控制窗口大小，还可以配合表格随意地在网页中的任何位置插入窗口。实际上就是在窗口中再创建一个窗口。

使用 iframe 创建浮动框架的格式如下：

```
<iframe src="链接对象" >
```

其中，src 表示浮动框架中显示对象的路径，可以是绝对路径，也可以是相对路径。例如，下面的代码是在浮动框架中显示百度网站。

【例 7.7】使用浮动框架(案例文件：ch07\7.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>浮动框架中显示百度网站</title>
</head>
<body>
<iframe src="http://www.baidu.com"></iframe>
</body>
</html>
```



图 7-14 浮动框架效果

在 IE 11.0 中预览网页，效果如图 7-14 所示。从预览结果可见，浮动框架在页面中又创建了一个窗口。在默认情况下，浮动框架的尺寸为 220 像素×120 像素。

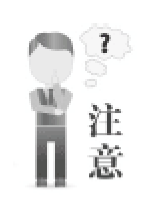
如果需要调整浮动框架尺寸，请使用 CSS 样式。修改上述浮动框架尺寸，请在 head 标记部分增加如下 CSS 代码：

```
<style>
iframe{
    width:600px;     //宽度
    height:800px;    //高度
    border:none;     //无边框
}
</style>
```

在 IE 11.0 中预览网页，效果如图 7-15 所示。



图 7-15 修改尺寸的浮动框架



在 HTML 5 中，iframe 仅支持 src 属性，再无其他属性。

7.4 综合案例——使用锚链接制作电子书阅读网页

超链接除了可以链接特定的文件和网站之外，还可以链接到网页内的特定内容。这可以使用<a>标签的 name 或 id 属性，创建一个文档内部的书签，也就是说，可以创建指向文档片段的链接。

例如，使用以下命令可以将网页中的文本“你好”定义为一个内部书签，书签名称为 name1。

```
<a name="name1" >你好</a>
```

在网页中的其他位置可以插入超链接引用该书签，引用命令如下：

```
<a href="#name1" >引用内部书签</a>
```

通常网页内容比较多的网站会采用这种方法，如一个电子书网页。

下面使用锚链接制作一个电子书网页。

step 01 新建记事本，输入以下代码，并保存为电子书.html 文件。

```
<!DOCTYPE html>
<html>
<head>
<title>电子书</title>
</head>
<body >
<h1>文学鉴赏</h1>
<ul>
   <li><a href="#第一篇" >再别康桥</a>
   <li><a href="#第二篇" >雨  巷</a>
   <li><a href="#第三篇" >荷塘月色</a>
</ul>
<h3><a name="第一篇" >再别康桥</a></h3>
<h3><a name="第二篇" >雨  巷</a></h3>
<h3><a name="第三篇" >荷塘月色</a></h3>
</body>
</html>
```

step 02 使用 IE 11.0 打开文件，显示效果如图 7-16 所示。



图 7-16 电子书网页

step 03 为每一个文学作品添加内容，完善后的代码如下。

```
<!DOCTYPE html>
<html>
<head>
<title>电子书</title>
</head>
<body >
<h1>文学鉴赏</h1>
<ul>
   <li><a href="#第一篇" >再别康桥</a>
   <li><a href="#第二篇" >雨  巷</a>
   <li><a href="#第三篇" >荷塘月色</a>
</ul>
<h3><a name="第一篇" >再别康桥</a></h3>
——徐志摩
<ul>
   <li>轻轻的我走了，正如我轻轻的来；
   <li>我轻轻的招手，作别西天的云彩。
     <br>
   <li>那河畔的金柳，是夕阳中的新娘；
   <li>波光里的艳影，在我的心头荡漾。
     <br>
   <li>软泥上的青荇，油油的在水底招摇；
   <li>在康河的柔波里，我甘心做一条水草！
     <br>
   <li>那榆荫下的一潭，不是清泉，是天上虹；
   <li>揉碎在浮藻间，沉淀着彩虹似的梦。
    <br>
   <li>寻梦？撑一支长篙，向青草更青处漫溯；
   <li>满载一船星辉，在星辉斑斓里放歌。
    <br>
   <li>但我不能放歌，悄悄是别离的笙箫；
   <li>夏虫也为我沉默，沉默是今晚的康桥！
    <br>
   <li>悄悄的我走了，正如我悄悄的来；
   <li>我挥一挥衣袖，不带走一片云彩。
</ul>
<h3><a name="第二篇" >雨  巷</a></h3>
——戴望舒<br>
撑着油纸伞，独自彷徨在悠长、悠长又寂寥的雨巷，我希望逢着一个丁香一样的结着愁怨的姑娘。
<br>
她是有丁香一样的颜色，丁香一样的芬芳，丁香一样的忧愁，在雨中哀怨，哀怨又彷徨；她彷徨在这寂
寥的雨巷，撑着油纸伞像我一样，像我一样地默默行着，冷漠，凄清，又惆怅。<br>
她静默地走近，走近，又投出太息一般的眼光，她飘过像梦一般地凄婉迷茫。像梦中飘过一枝丁香的，
我身旁飘过这女郎；她静默地远了，远了，到了颓圮的篱墙，走尽这雨巷。在雨的哀曲里，消了她的颜
色，散了她的芬芳，消散了，甚至她的太息般的眼光丁香般的惆怅。撑着油纸伞，独自彷徨在悠长，悠
长又寂寥的雨巷，我希望飘过一个丁香一样的结着愁怨的姑娘。
<h3><a name="第三篇" >荷塘月色</a></h3>
曲曲折折的荷塘上面，弥望的是田田的叶子。叶子出水很高，像亭亭的舞女的裙。层层的叶子中间，零
星地点缀着些白花，有袅娜地开着的，有羞涩地打着朵儿的；正如一粒粒的明珠，又如碧天里的星星，
又如刚出浴的美人。微风过处，送来缕缕清香，仿佛远处高楼上渺茫的歌声似的。这时候叶子与花也有
一丝的颤动，像闪电般，霎时传过荷塘的那边去了。叶子本是肩并肩密密地挨着，这便宛然有了一道凝
碧的波痕。叶子底下是脉脉的流水，遮住了，不能见一些颜色；而叶子却更见风致了。<br>
```

```
月光如流水一般，静静地泻在这一片叶子和花上。薄薄的青雾浮起在荷塘里。叶子和花仿佛在牛乳中洗过一样；又像笼着轻纱的梦。虽然是满月，天上却有一层淡淡的云，所以不能朗照；但我以为这恰是到了好处——酣眠固不可少，小睡也别有风味的。月光是隔了树照过来的，高处丛生的灌木，落下参差的斑驳的黑影，峭楞楞如鬼一般；弯弯的杨柳的稀疏的倩影，却又像是画在荷叶上。塘中的月色并不均匀；但光与影有着和谐的旋律，如梵婀玲上奏着的名曲。
</body>
</html>
```

step 04 保存文件，使用 IE 11.0 打开文件，效果如图 7-17 所示。



图 7-17 添加网页内容

step 05 单击“雨巷”超链接，页面会自动跳转到“雨巷”对应的内容，如图 7-18 所示。

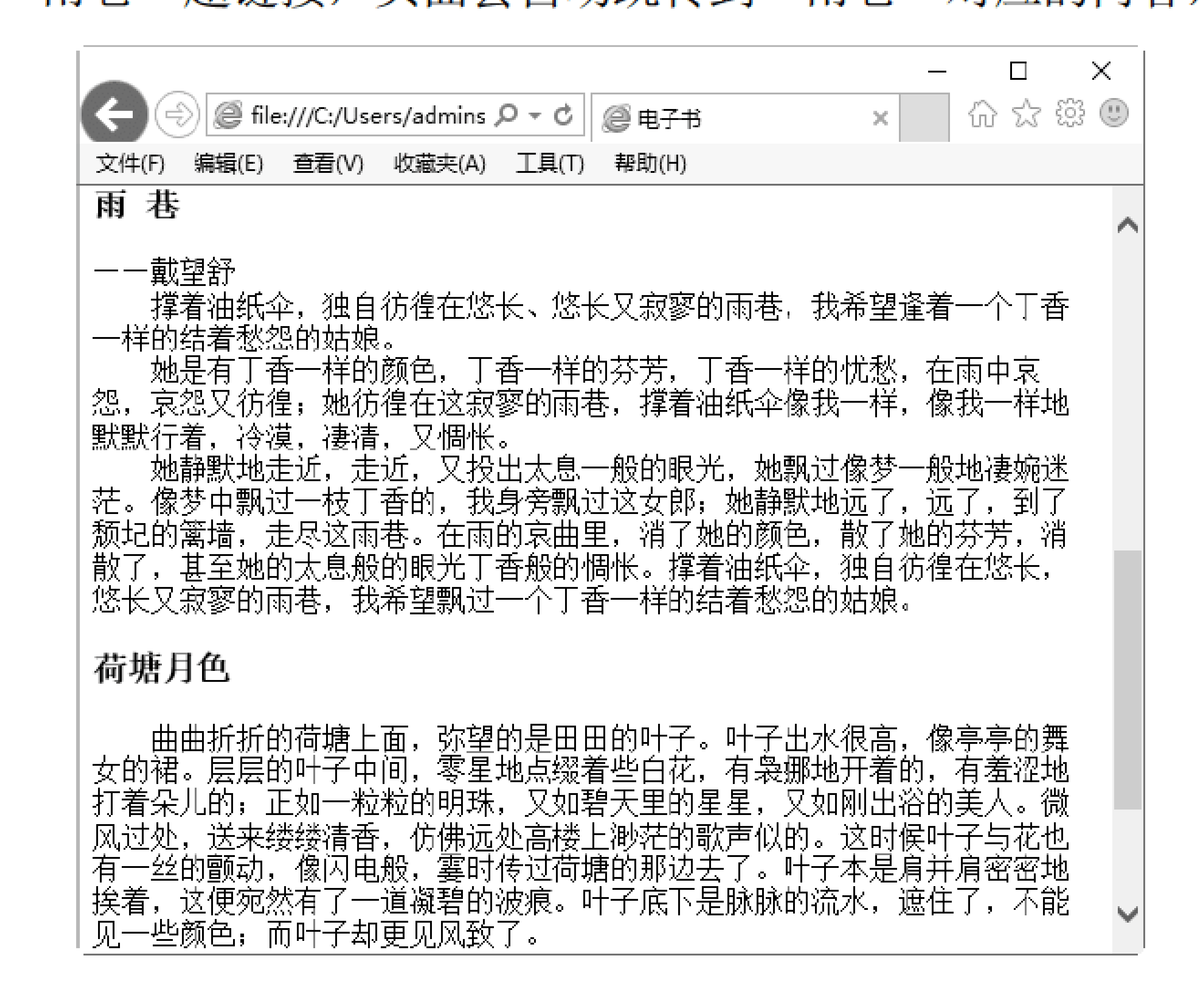


图 7-18 网页效果

7.5 高手解惑

疑问 1：在创建超链接时，使用绝对 URL 还是相对 URL？

答：在创建超链接时，如果要链接的是另外一个网站中的资源，需要使用完整的绝对 URL；如果在网页中创建内部链接，一般使用相对于当前文档或站点根文件夹的相对 URL。

疑问 2：链接增多后的网站，如何设置目录结构以方便维护？

答：当一个网站的网页数量增加到一定程度以后，网站的管理与维护将变得非常烦琐。因此，掌握一些网站管理与维护的技术是非常实用的，可以节省很多时间。建立适合的网站文件存储结构，可以方便网站的管理与维护。通常使用的 3 种网站文件组织结构方案及文件管理遵循的原则如下。

(1) 按照文件的类型进行分类管理。将不同类型的文件放在不同的文件夹中，这种存储方法适合于中小型网站，这种方法是通过文件的类型对文件进行管理。

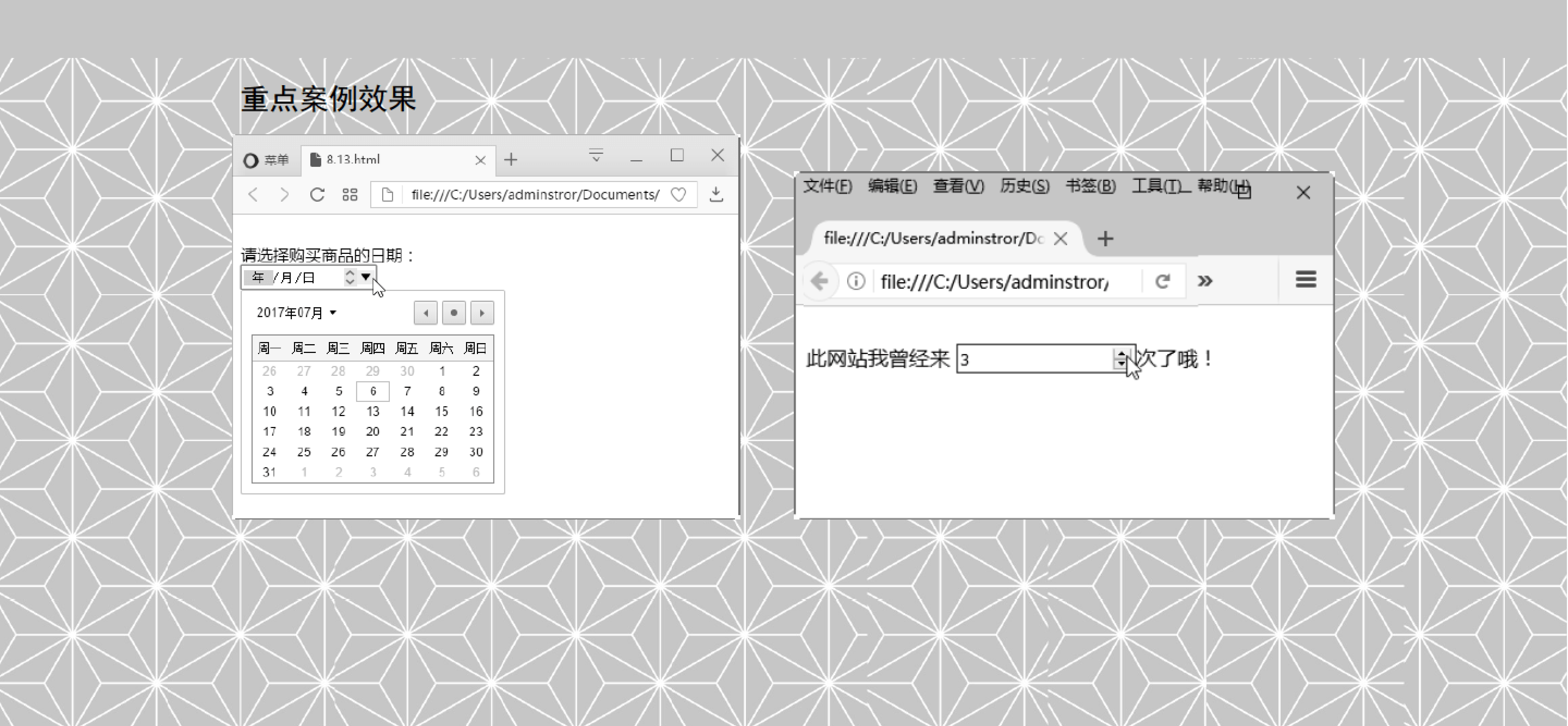
(2) 按照主题对文件进行分类。网站的页面按照不同的主题进行分类储存。同一主题的所有文件存放在一个文件夹中，然后再进一步细分文件的类型。这种方案适用于页面和文件数量众多、信息量大的静态网站。

(3) 对文件类型进行进一步细分存储管理。这种方案是第一种存储方案的深化，将页面进一步细分后进行分类存储管理。这种方案适用于文件类型复杂、包含各种文件的多媒体动态网站。

第 8 章

使用 HTML 5 创建表单

在网页中，表单的作用比较重要，主要负责采集浏览者的相关数据。例如常见的登录表、调查表、留言表等。在 HTML 5 中，表单拥有多个新的表单输入类型，这些新特性提供了更好的输入控制和验证。



8.1　案例 1——认识表单

表单主要用于收集网页上浏览者的相关信息。其标签为<form></form>。表单的基本语法格式如下：

```
<form action="url" method="get|post" enctype="mime"></form>
```

其中，action="url"指定处理提交表单的格式，它可以是一个 URL 地址或一个电子邮件地址。method="get"或"post"指明提交表单的 HTTP 方法。enctype="mime"指明用来把表单提交给服务器时的互联网媒体形式。

表单是一个能够包含表单元素的区域。通过添加不同的表单元素，将显示不同的效果。

【例 8.1】认识表单(案例文件：ch08\8.1.html)。

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<form>
下面是输入用户登录信息
<br>
用户名称
<input type="text" name="user">
    <br>
    用户密码
    <input type="password" name="password"><br>
    <input type="submit" value="登录">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-1 所示，可以看到用户登录信息页面。

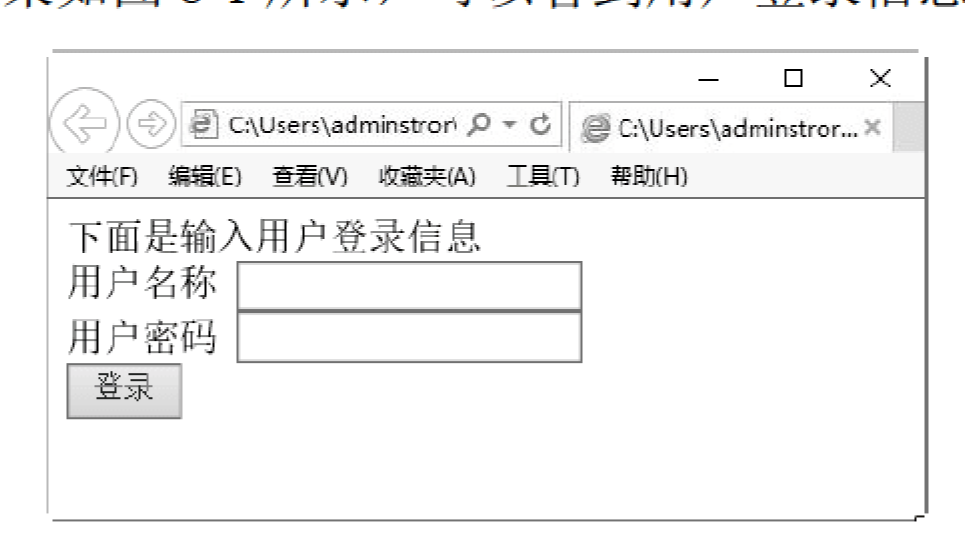


图 8-1　用户登录窗口

8.2　表单基本元素的使用

表单元素是能够让用户在表单中输入信息的元素。常见的有文本框、单选按钮、复选框、列表框等。下面主要讲解表单基本元素的使用方法和技巧。

8.2.1　案例 2——单行文本输入框 text

文本框是一种让访问者自己输入内容的表单对象，通常被用来填写单个字或者简短的回答，如用户姓名和地址等。

代码格式如下：

```
<input type="text" name="..." size="..." maxlength="..." value="...">
```

其中，type="text"定义单行文本输入框。name 属性定义文本框的名称，要保证数据的准确采集，必须定义一个独一无二的名称。size 属性定义文本框的宽度，单位是单个字符宽度。maxlength 属性定义最多输入的字符数。value 属性定义文本框的初始值。

【例 8.2】使用单行文本输入框(案例文件：ch08\8.2.html)。

```
<!DOCTYPE html>
<html>
<head><title>输入用户的姓名</title></head>
<body>
<form>
    请输入您的姓名：
    <input type="text" name="yourname" size="20" maxlength="15">
    请输入您的地址：
    <input type="text" name="youradr" size="20" maxlength="15">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-2 所示，可以看到两个单行文本输入框。



图 8-2　单行文本输入框

8.2.2　案例 3——多行文本输入框 textarea

多行文本输入框(textarea)主要用于输入较长的文本信息。其代码格式如下：

```
<textarea name="..." cols="..." rows="..." wrap="..."></textarea>
```

其中，name 属性定义多行文本框的名称，要保证数据的准确采集，必须定义一个独一无二的名称。cols 属性定义多行文本框的宽度，单位是单个字符宽度。rows 属性定义多行文本框的高度，单位是单个字符宽度。wrap 属性定义输入内容大于文本域时显示的方式。

【例 8.3】使用多行文本输入框(案例文件：ch08\8.3.html)。

```
<!DOCTYPE html>
<html>
<head><title>多行文本输入</title></head>
<body>
<form>
   请输入您最新的工作情况<br>
   <textarea name="yourworks" cols ="50" rows = "5"></textarea>
   <br>
   <input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-3 所示，可以看到多行文本输入框。

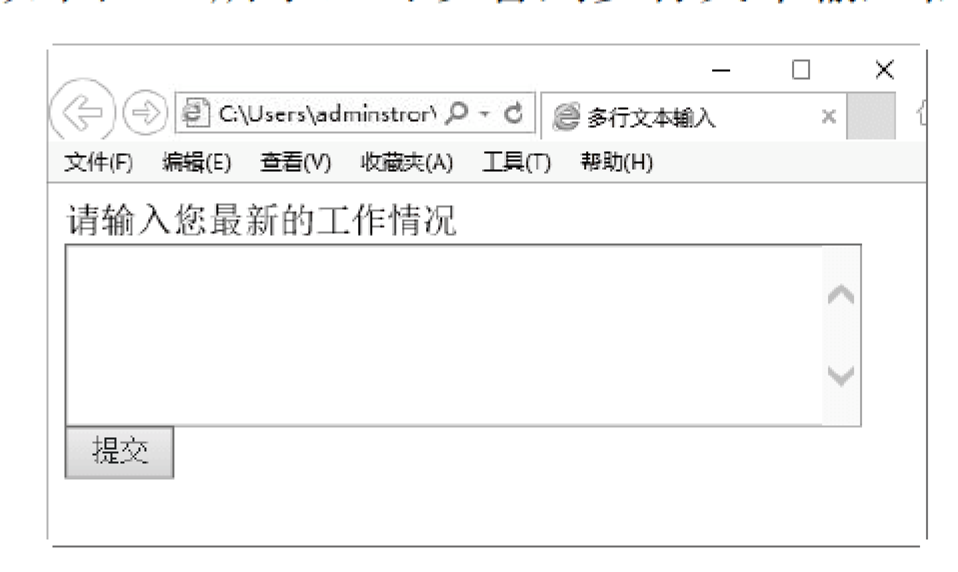


图 8-3　多行文本输入框

8.2.3　案例 4——密码域 password

密码输入框是一种特殊的文本域，主要用于输入一些保密信息。当网页浏览者输入文本时，显示的是黑点或者其他符号，这样就增加了输入文本的安全性。其代码格式如下：

```
<input type="password" name="..." size="..." maxlength="...">
```

其中 type="password"定义密码框。name 属性定义密码框的名称，要保证唯一性。size 属性定义密码框的宽度，单位是单个字符宽度。maxlength 属性定义最多输入的字符数。

【例 8.4】使用密码域(案例文件：ch08\8.4.html)。

```
<!DOCTYPE html>
<html>
<head><title>输入用户姓名和密码</title></head>
<body>
<form>
用户姓名：
<input type="text" name="yourname">
<br>
登录密码：
<input type="password" name="yourpw"><br>
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-4 所示。输入用户名和密码时，可以看到密码以黑点的形式显示。如果想查看输入的登录密码，可以单击眼睛图标，即可显示输入的具体密码。



图 8-4　密码输入框

8.2.4　案例 5——单选按钮 radio

单选按钮主要是让网页浏览者在一组选项里只能选择一个。其代码格式如下：

```
<input type="radio" name="" value="">
```

其中，type="radio"定义单选按钮。name 属性定义单选按钮的名称，单选按钮都是以组为单位使用的，在同一组中的单选按钮都必须用同一个名称。value 属性定义单选按钮的值，在同一组中，它们的域值必须是不同的。

【例 8.5】使用单选按钮(案例文件：ch08\8.5.html)。

```
<!DOCTYPE html>
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form>
    请选择您感兴趣的图书类型：
    <br>
    <input type="radio" name="book" value="Book1">网站编程<br>
    <input type="radio" name="book" value="Book2">办公软件<br>
    <input type="radio" name="book" value="Book3">设计软件<br>
    <input type="radio" name="book" value="Book4">网络管理<br>
    <input type="radio" name="book" value="Book5">黑客攻防<br>
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-5 所示。可以看到 5 个单选按钮，而用户只能选中其中一个单选按钮。

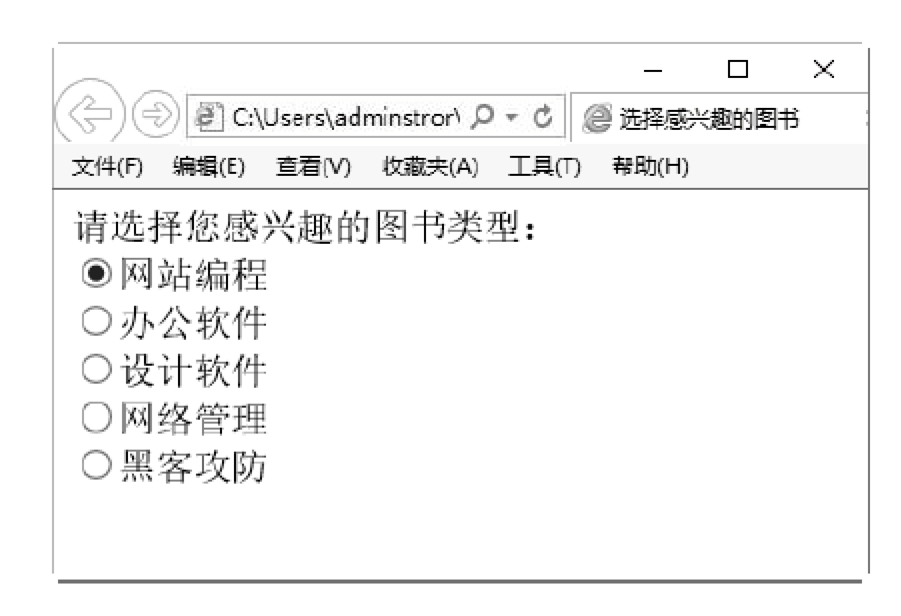


图 8-5　单选按钮

8.2.5 案例 6——复选框 checkbox

复选框主要是让网页浏览者在一组选项里可以同时选择多个选项。每个复选框都是一个独立的元素，都必须有一个唯一的名称。其代码格式如下：

```
<input type="checkbox" name="" value="">
```

其中 type="checkbox"定义复选框。name 属性定义复选框的名称，在同一组中的复选框都必须用同一个名称。value 属性定义复选框的值。

【例 8.6】使用复选框(案例文件：ch08\8.6.html)。

```
<!DOCTYPE html>
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form>
    请选择您感兴趣的图书类型：<br>
    <input type="checkbox" name="book" value="Book1">网站编程<br>
    <input type="checkbox" name="book" value="Book2">办公软件<br>
    <input type="checkbox" name="book" value="Book3">设计软件<br>
    <input type="checkbox" name="book" value="Book4">网络管理<br>
    <input type="checkbox" name="book" value="Book5" checked>黑客攻防<br>
</form>
</body>
</html>
```



checked 属性主要用来设置默认选中项。

在 IE 11.0 中浏览，可以看到 5 个复选框，其中“黑客攻防”复选框被默认勾选。同时，浏览者还可以勾选其他复选框。效果如图 8-6 所示。

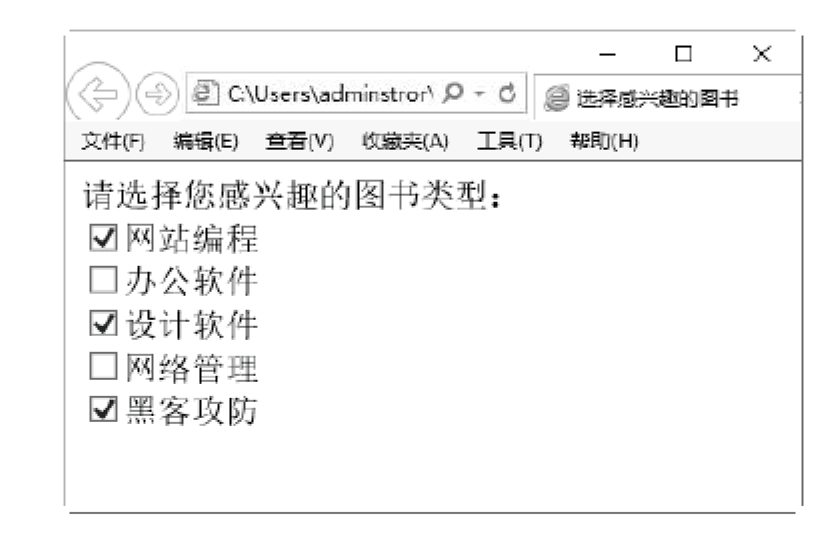


图 8-6 复选框的效果

8.2.6 案例 7——列表框 select

列表框主要用于在有限的空间里设置多个选项。列表框既可以用作单选，也可以用作复选。其代码格式如下：

```
<select name="..." size="..." multiple>
    <option value="..." selected>
      ...
    </option>
    ...
</select>
```

其中 size 属性定义列表框的行数。name 属性定义列表框的名称。multiple 属性表示可以多选，如果不设置本属性，那么只能单选。value 属性定义列表项的值。selected 属性表示默认已经选中本选项。

【例 8.7】使用列表框(案例文件：ch08\8.7.html)。

```
<!DOCTYPE html>
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form>
    请选择您感兴趣的图书类型：<br>
    <select name="fruit" size = "3" multiple>
        <option value="Book1">网站编程
        <option value="Book2">办公软件
        <option value="Book3">设计软件
        <option value="Book4">网络管理
        <option value="Book5">黑客攻防
    </select>
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-7 所示。可以看到列表框，其中显示了 3 行选项，用户可以按住 Ctrl 键，选择多个选项。



图 8-7　列表框的效果

8.2.7　案例 8——普通按钮 button

普通按钮用来控制其他定义了处理脚本的处理工作。其代码格式如下：

```
<input type="button" name="..." value="..." onClick="...">
```

其中 type="button"定义为普通按钮。name 属性定义普通按钮的名称。value 属性定义按钮的显示文字。onClick 属性表示单击行为，也可以是其他事件，通过指定脚本函数来定义按钮的行为。

【例 8.8】使用普通按钮(案例文件：ch08\8.8.html)。

```
<!DOCTYPE html>
<html/>
<body/>
<form/>
点击下面的按钮，把文本框 1 的内容拷贝到文本框 2 中：
<br>
文本框 1：<input type="text" id="field1" value="学习 HTML 5 的技巧">
```

```
    <br>
    文本框 2: <input type="text" id="field2">
    <br>
    <input type="button" name="..." value="单击我" onClick="document
      .getElementById('field2').value=document
      .getElementById('field1').value">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-8 所示，单击“单击我”按钮，即可实现将文本框 1 中的内容复制到文本框 2 中的操作。

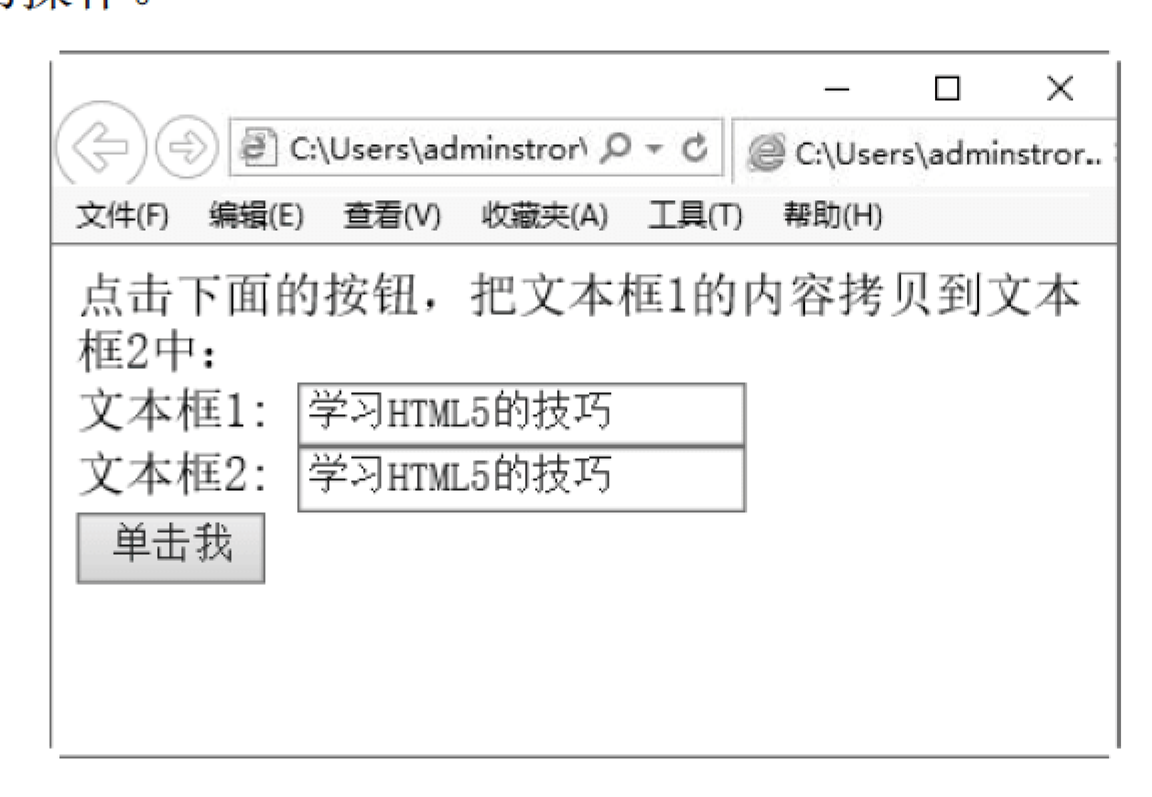


图 8-8 单击“单击我”按钮后的复制效果

8.2.8 案例 9——提交按钮 submit

提交按钮用来将输入的信息提交到服务器。其代码格式如下：

```
<input type="submit" name="..." value="...">
```

其中 type="submit"定义为提交按钮。name 属性定义提交按钮的名称。value 属性定义按钮的显示文字。通过提交按钮，可以将表单里的信息提交给表单中 action 所指向的文件。

【例 8.9】使用提交按钮(案例文件：ch08\8.9.html)。

```
<!DOCTYPE html>
<html>
<head><title>输入用户名信息</title></head>
<body>
<form  action="http://www.yinhangit.com/yonghu.asp" method="get">
    请输入你的姓名：
    <input type="text" name="yourname">
    <br>
    请输入你的住址：
    <input type="text" name="youradr">
    <br>
    请输入你的单位：
    <input type="text" name="yourcom">
    <br>
    请输入你的联系方式：
```

```
    <input type="text" name="yourcom">
    <br>
    <input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-9 所示，输入内容后单击“提交”按钮，即可将表单中的数据发送到指定的文件。

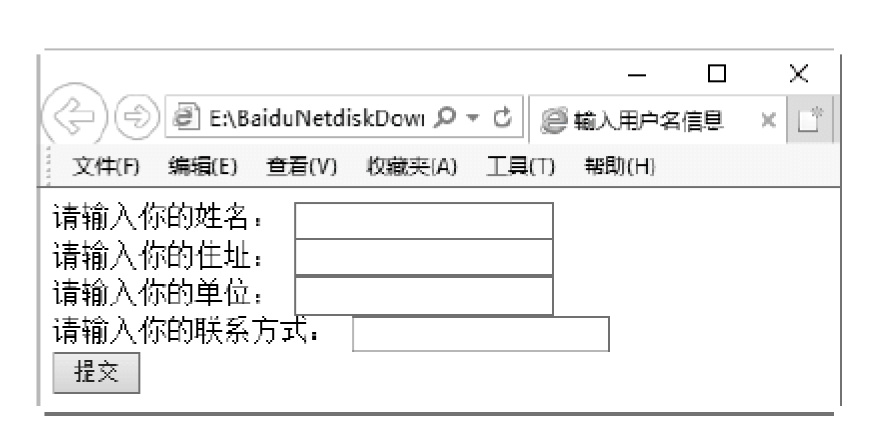


图 8-9　提交按钮

8.2.9　案例 10——重置按钮 reset

重置按钮又称为复位按钮，用来重置表单中输入的信息。其代码格式如下：

```
<input type="reset" name="..." value="...">
```

其中，type="reset"定义复位按钮。name 属性定义复位按钮的名称。value 属性定义按钮的显示文字。

【例 8.10】使用重置按钮(案例文件：ch08\8.10.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
请输入用户名称:
<input type='text'>
<br/>
请输入用户密码:
    <input type='password'>
    <br>
    <input type="submit" value="登录">
    <input type="reset" value="重置">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-10 所示，输入内容后单击“重置”按钮，即可实现将表单中的数据清空的目的。

图 8-10　重置按钮

8.3 表单高级元素的使用

除了上述基本元素外，HTML 5 中还有一些高级元素，包括 url、email、time、number、range、required 等。

8.3.1 案例 11——url 属性的使用

url 属性是用于说明网站网址的。显示为一个文本字段输入 URL 地址。在提交表单时，会自动验证 url 的值。其代码格式如下：

```
<input type="url" name="userurl"/>
```

另外，用户可以使用普通属性设置 url 输入框。例如，可以使用 max 属性设置其最大值、min 属性设置其最小值、step 属性设置合法的数字间隔，利用 value 属性规定其默认值。对于另外的高级属性中同样的设置不再重复讲述。

【例 8.11】 url 属性的使用(案例文件：ch08\8.11.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
    <br/>
    请输入网址：
    <input type="url" name="userurl"/>
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-11 所示，用户即可输入相应的网址。



图 8-11 url 属性的效果

8.3.2 案例 12——email 属性的使用

与 url 属性类似，email 属性用于让浏览者输入 E-mail 地址。在提交表单时，会自动验证 email 域的值。其代码格式如下：

```
<input type="email" name="user_email"/>
```

【例 8.12】email 属性的使用(案例文件：ch08\8.12.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
   <br/>
   请输入您的邮箱地址：
   <input type="email" name="user email"/>
   <br>
   <input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-12 所示，用户即可输入相应的邮箱地址。如果用户输入的邮箱地址不合法，单击“提交”按钮后，会弹出提示信息。



图 8-12 email 属性的效果

8.3.3 案例 13——date 和 time 属性的使用

在 HTML 5 中，新增了一些日期和时间输入类型，包括 date、datetime、datetime-local、month、week 和 time。它们的具体含义如表 8-1 所示。

表 8-1 HTML 5 中新增的一些日期和时间属性

属　性	含　义
date	选取日、月、年
month	选取月、年
week	选取周和年
time	选取时间
datetime	选取时间、日、月、年
datetime-local	选取时间、日、月、年(本地时间)

上述属性的代码格式彼此类似，以 date 属性为例，其代码格式如下：

```
<input type="date" name="user_date" />
```

【例 8.13】date 属性的使用(案例文件：ch08\8.13.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
    <br/>
    请选择购买商品的日期：
    <br>
    <input type="date" name="user date"/>
</form>
</body>
</html>
```

在 Opera 45.0 中浏览，效果如图 8-13 所示，用户单击输入框中的向下按钮，即可在弹出的窗口中选择需要的日期。

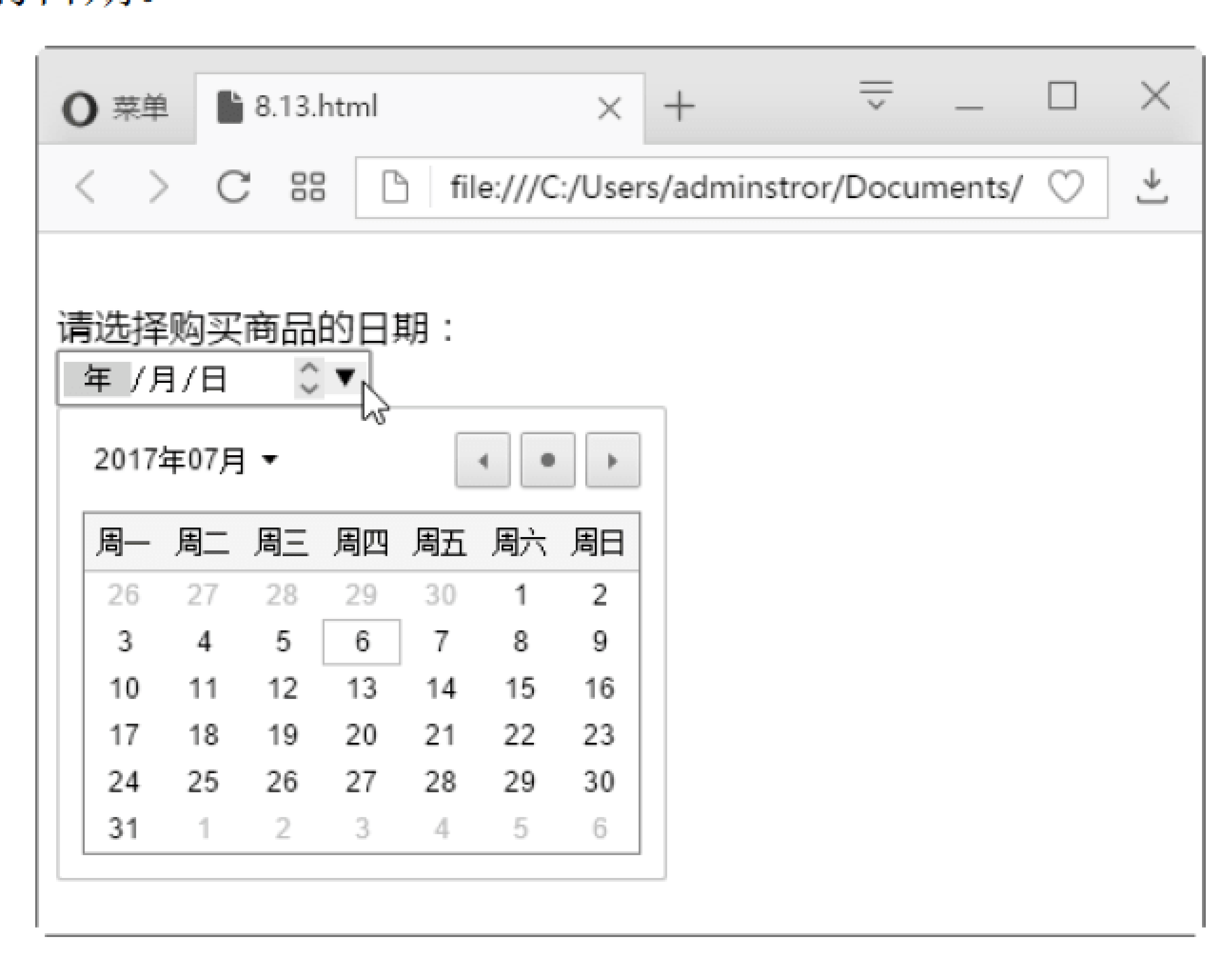


图 8-13　date 属性的效果

8.3.4　案例 14——number 属性的使用

number 属性提供了一个输入数字的输入类型。用户可以直接输入数值，或者通过单击微调框中的向上或者向下按钮来选择数值。代码格式如下：

```
<input type="number" name="shuzi" />
```

【例 8.14】number 属性的使用(案例文件：ch08\8.14.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
    <br/>
    此网站我曾经来
    <input type="number" name="shuzi"/>次了哦!
</form>
```

```
</body>
</html>
```

在 Firefox 53.0 中浏览，效果如图 8-14 所示，用户可以直接输入数值，也可以单击微调框中的向上或者向下按钮选择合适的数值。



图 8-14　number 属性的效果



强烈建议用户使用 min 和 max 属性规定输入的最小值和最大值。

8.3.5　案例 15——range 属性的使用

range 属性显示为一个滑条控件。与 number 属性一样，用户可以使用 max、min 和 step 属性来控制控件的范围。其代码格式如下：

```
<input type="range" name="" min="" max="" />
```

其中 min 和 max 分别控制滑条控件的最小值和最大值。

【例 8.15】range 属性的使用(案例文件：ch08\8.15.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
    <br/>
    英语成绩公布了！我的成绩名次为：
    <input type="range" name="ran" min="1" max="10"/>
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-15 所示，用户可以拖曳滑块，从而选择合适的数值。



在默认情况下，滑块位于中间位置。如果用户指定的最大值小于最小值，则允许使用反向滑条，目前浏览器对这一属性还不能很好地支持。

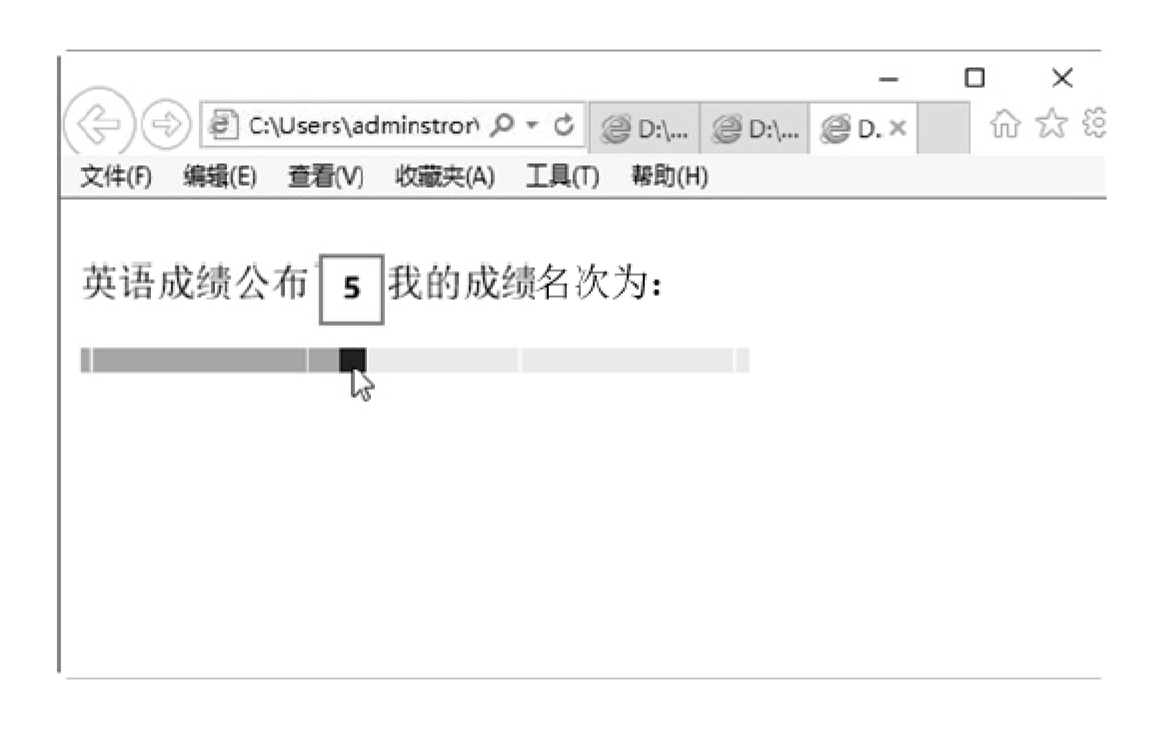


图 8-15　range 属性的效果

8.3.6　案例 16——required 属性的使用

required 属性规定必须在提交之前填写输入域(不能为空)。

required 属性适用于以下类型的输入属性：text、search、url、email、password、date、pickers、number、checkbox、radio 等。

【例 8.16】required 属性的使用(案例文件：ch08\8.16.html)。

```
<!DOCTYPE html>
<html>
<body>
<form>
    下面是输入用户登录信息
    <br>
    用户名称
    <input type="text" name="user" required="required">
    <br>
    用户密码
    <input type="password" name="password" required="required">
    <br>
    <input type="submit" value="登录">
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-16 所示。用户如果只是输入密码，然后单击“登录”按钮，将弹出提示信息。



图 8-16　required 属性的效果

8.4 综合案例——创建用户反馈表单

本案例将使用一个表单内的各种元素来开发一个简单的网站用户意见反馈页面。

具体操作步骤如下。

step 01 分析需求。反馈表单非常简单，通常包含 3 个部分，需要在页面上方给出标题，标题下方是 nl 正文部分，即表单元素，最下方是表单元素提交按钮。在设计这个页面时，需要把“用户注册”标题设置成 h1 大小，正文使用 p 来限制表单元素。

step 02 构建 HTML 页面，实现表单内容：

```
<!DOCTYPE html>
<html>
<head>
<title>用户反馈页面</title>
</head>
<body>
<h1 align=center>用户反馈表单</h1>
<form method="post">
    <p>姓    名：
    <input type="text" class="txt" size="12"
      maxlength="20" name="username" /></p>
    <p>性    别：
    <input type="radio" value="male" />男
    <input type="radio" value="female" />女</p>
    <p>年    龄：
    <input type="text" class=txt name="age" /></p>
    <p>联系电话：
    <input type="text" class=txt name="tel" /></p>
    <p>电子邮件：
    <input type="text" class=txt name="email" /></p>
    <p>联系地址：
    <input type="text" class=txt name="address" /></p>
    <p>
    请输入您对网站的建议<br>
    <textarea name="yourworks" cols ="50" rows = "5"></textarea>
    <br>
    <input type="submit" name="submit" value="提交"/>
    <input type="reset" name="reset" value="清除" /></p>
</form>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 8-17 所示。可以看到，创建了一个用户反馈表单，包含标题以及“姓名”“性别”“年龄”“联系电话”“电子邮件”“联系地址”“请输入您对网站的建议”等输入框和“提交”按钮等。

图 8-17　用户反馈页面

8.5　跟我学上机——制作用户注册表单

注册表单非常简单，通常包含 3 个部分，需要在页面上方给出标题，标题下方是正文部分，即表单元素，最下方是表单元素提交按钮。具体操作步骤如下。

step 01 打开记事本文件，在其中输入下述代码：

```
<!DOCTYPE html>
<html>
<head>
<title>注册表单</title>
</head>
<body>
<h1 align=center>用户注册</h1>
<form method="post">
    <p>姓    名:
    <input type="text" class=txt size="12"
      maxlength="20" name="username" />
    </p>
    <p>性    别:
    <input type="radio" value="male" />男
    <input type="radio" value="female" />女
    </p>
    <p>年    龄:
    <input type="text" class=txt name="age" />
    </p>
    <p>联系电话:
    <input type="text" class=txt name="tel" />
    </p>
```

```
    <p>电子邮件：
    <input type="text" class=txt name="email" />
    </p>
    <p>联系地址：
    <input type="text"  class=txt name="address" />
    </p>
    <p>
    <input type="submit" name="submit" value="提交" class=but />
    <input type="reset" name="reset" value="清除" class=but  />
    </p>
</form>
</body>
</html>
```

step 02 保存网页，在 IE 11.0 中预览，效果如图 8-18 所示。

图 8-18 网页预览效果

8.6 高 手 解 惑

疑问 1：如何在表单中实现文件上传框？

答：在 HTML 5 语言中，使用 file 属性实现文件上传框。语法格式为：<input type="file" name="..."size=" "maxlength=" ">。其中 type="file"定义为文件上传框；name 属性定义文件上传框的名称；size 属性定义文件上传框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数。文件上传框的显示效果如图 8-19 所示。



图 8-19　文件上传框

疑问 2：制作的单选按钮为什么可以同时选中多个？

答：此时用户需要检查单选按钮的名称，保证同一组中的单选按钮名称必须相同，这样才能保证单选按钮只能选中其中一个。

第 9 章 使用 HTML 5 创建表格

HTML 中的表格不但可以清晰地显示数据，而且可以用于页面布局。HTML 中的表格类似于 Word 软件中的表格，尤其是使用网页制作工具，操作很相似。

HTML 表格制作是使用相关标记(如表格对象 table 标记、行对象 tr 标记、单元格对象 td 标记)来完成的。

重点案例效果



9.1　案例 1——表格的基本结构

使用表格显示数据，可以更直观和清晰。在 HTML 文档中，表格主要用于显示数据，虽然可以使用表格布局，但是不建议使用，它有很多弊端。表格一般由行、列和单元格组成，如图 9-1 所示。

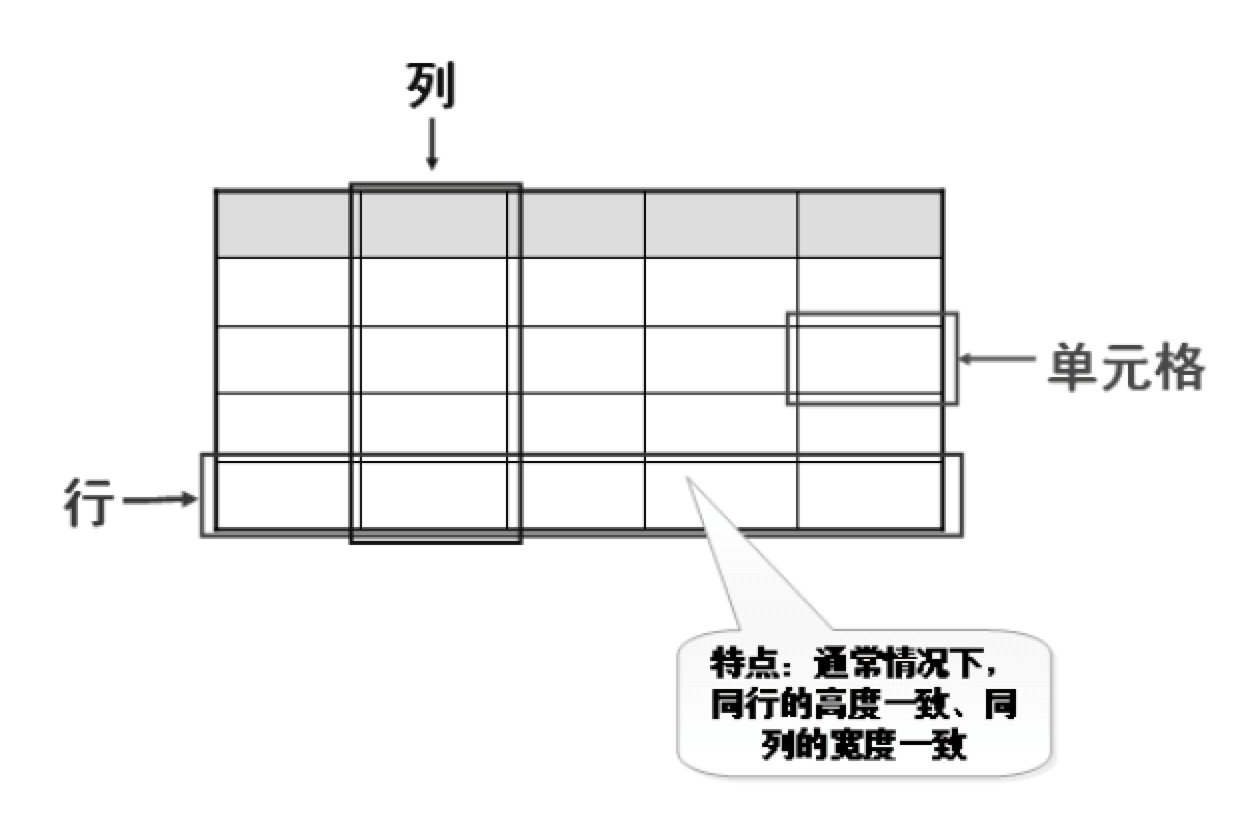


图 9-1　表格的组成

在 HTML 5 中，用于创建表格的标记如下。

(1) <table>。用于标识一个表格对象的开始，</table>标记标识一个表格对象的结束。一个表格中，只允许出现一对<table>标记。HTML 5 中不再支持它的任何属性。

(2) <tr>。用于标识表格一行的开始，</tr>标记用于标识表格一行的结束。表格内有多少对<tr></tr>标记，就表示表格中有多少行。HTML 5 中不再支持它的任何属性。

(3) <td>。用于标识表格某行中的一个单元格的开始，</td>标记用于标识表格某行中一个单元格的结束。<td></td>标记应书写在<tr></tr>标记内，一对<tr></tr>标记内有多少对<td></td>标记，就表示该行有多少个单元格。HTML 5 中，<td>仅有 colspan 和 rowspan 两个属性。

最基本的表格，必须包含一对<table></table>标记、一对或几对<tr></tr>标记以及一对或几对<td></td>标记。一对<table></table>标记定义一个表格，一对<tr></tr>标记定义一行，一对<td></td>标记定义一个单元格。

【例 9.1】定义一个 4 行 3 列的表格(案例文件：ch09\9.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>表格基本结构</title>
</head>
<body>
<table border="1">
  <tr>
    <td>A1</td>
    <td>B1</td>
```

```
    <td>C1</td>
  </tr>
  <tr>
    <td>A2</td>
    <td>B2</td>
    <td>C2</td>
  </tr>
  <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
  </tr>
  <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 9-2 所示。



图 9-2　定义一个 4 行 3 列的表格



从预览图中，读者会发现，表格没有边框，行高及列宽也无法控制。进行上述知识讲述时，提到 HTML 5 中除了 td 标记提供两个单元格合并属性之外，<table>和<tr>标记也没有任何属性。

9.2　创 建 表 格

表格可以分为普通表格以及带有标题的表格，在 HTML 5 中，可以创建这两种表格。

9.2.1　案例 2——创建普通表格

例如创建 1 列、1 行 3 列和 2 行 3 列的三个表格。

【例 9.2】创建普通表格(案例文件：ch09\9.2.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>一列：</h4>
<table border="1">
<tr>
  <td>100</td>
</tr>
</table>
<h4>一行三列：</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
</table>
<h4>两行三列：</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 9-3 所示。

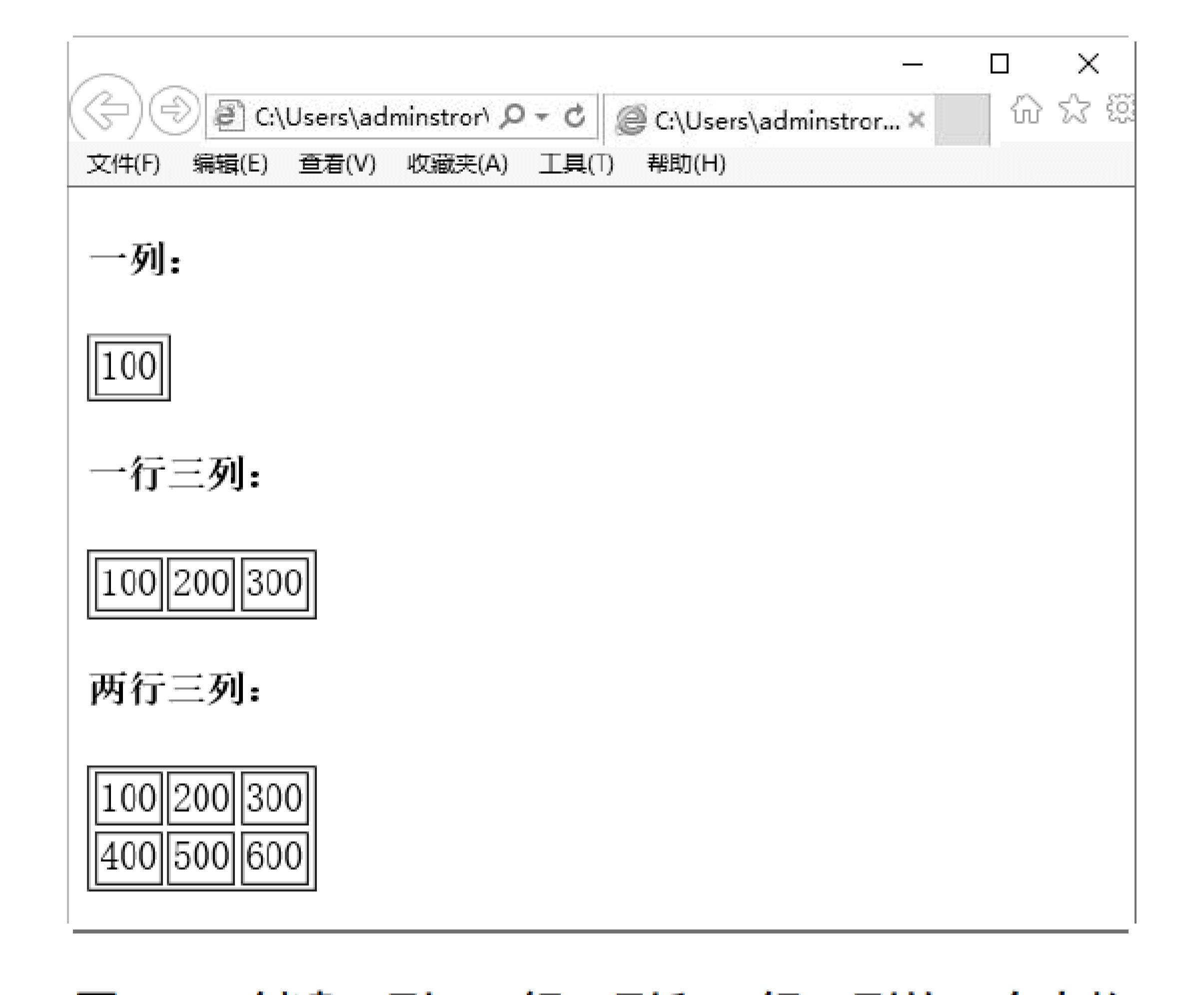


图 9-3　创建 1 列、1 行 3 列和 2 行 3 列的三个表格

9.2.2　案例 3——创建一个带有标题的表格

有时，为了方便表述表格，还需要在表格的上面加上标题。

【例 9.3】创建一个带有标题的表格(案例文件：ch09\9.3.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>带有标题的表格</h4>
<table border="3">
<caption>数据统计表</caption>
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
```

```
  <td>500</td>
  <td>600</td>
</tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 9-4 所示。



图 9-4　创建一个带有标题的表格

9.3　编 辑 表 格

在创建好表格之后，还可以编辑表格，包括设置表格的边框类型、设置表格的表头、合并单元格等。

9.3.1　案例 4——定义表格的边框类型

使用表格的 border 属性可以定义表格的边框类型，如常见的加粗边框的表格。

【例 9.4】创建不同边框类型的表格(案例文件：ch09\9.4.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>普通边框</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
<h4>加粗边框</h4>
<table border="8">
<tr>
  <td>First</td>
```

```
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 9-5 所示。



图 9-5　创建不同边框类型的表格

9.3.2　案例 5——定义表格的表头

表格中也存在有表头，常见的表头分为水平的和垂直的两种。例如，分别创建带有水平和垂直表头的表格。

【例 9.5】定义表格的表头(案例文件：ch09\9.5.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>水平的表头</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <th>性别</th>
  <th>电话</th>
</tr>
<tr>
  <td>张三</td>
  <td>男</td>
  <td>123456</td>
</tr>
</table>
<h4>垂直的表头：</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <td>小丽</td>
```

```
</tr>
<tr>
  <th>性别</th>
  <td>女</td>
</tr>
<tr>
  <th>电话</th>
  <td>123456</td>
</tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页，效果如图 9-6 所示。



姓名	性别	电话
张三	男	123456

姓名	小丽
性别	女
电话	123456

图 9-6　分别创建带有水平和垂直表头的表格

9.3.3　案例 6——设置表格背景

当创建好表格后，为了美观，还可以设置表格的背景，如为表格定义背景颜色、为表格定义背景图片等。

1. 定义表格背景颜色

为表格添加背景颜色是美化表格的一种方式。

【例 9.6】为表格添加背景颜色(案例文件：ch09\9.6.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>背景颜色：</h4>
<table border="1"
bgcolor="green">
<tr>
  <td>100</td>
  <td>200</td>
</tr>
<tr>
  <td>300</td>
  <td>400</td>
</tr>
</table>
</body>
</html>
```



图 9-7　为表格添加背景颜色

在 IE 11.0 中预览网页，效果如图 9-7 所示。

2. 定义表格背景图片

除了可以为表格添加背景颜色外，还可以将图片设置为表格的背景。例如，为表格添加背景图片。

【例 9.7】定义表格背景图片(案例文件：ch09\9.7.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>背景图片：</h4>
<table border="1" background="images/1.gif">
<tr>
  <td>100</td>
  <td>200</td>
</tr>
<tr>
  <td>300</td>
  <td>400</td>
</tr>
</table>
</body>
</html>
```



图 9-8　为表格添加背景图片

在 IE 11.0 中预览网页，效果如图 9-8 所示。

9.3.4　案例 7——设置单元格的背景

除了可以为表格设置背景外，还可以为单元格设置背景，包括添加背景颜色和背景图片两种。

【例 9.8】为单元格添加背景(案例文件：ch09\9.8.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>单元格背景</h4>
<table border="1">
<tr>
  <td bgcolor="red">100000</td>
  <td>200000</td>
</tr>
<tr>
  <td background="images/1.gif">200000</td>
  <td>300000</td>
</tr>
</table>
</body>
</html>
```

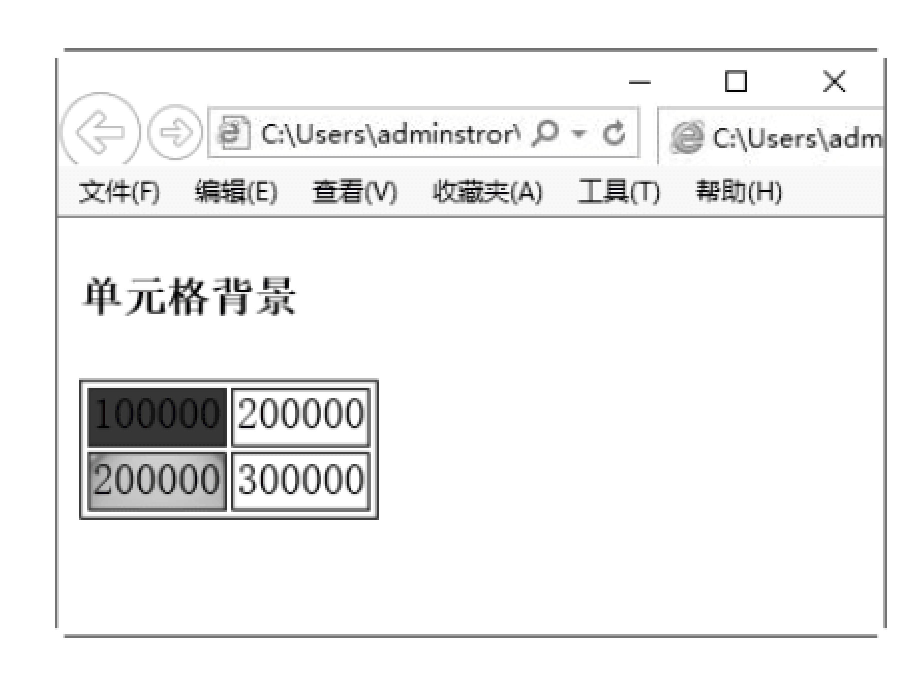


图 9-9　为单元格添加背景

在 IE 11.0 中预览网页，效果如图 9-9 所示。

9.3.5　案例 8——合并单元格

在实际应用中，并非所有表格都是规范的几行几列，而是需要将某些单元格进行合并，以符合某种内容上的需要。在 HTML 中，合并的方向有两种：一是上下合并；二是左右合并。这两种合并方式只需要使用 td 标记的两个属性即可。

1. 用 colspan 属性合并左右单元格

左右单元格的合并需要使用 td 标记的 colspan 属性来完成，其语法格式如下：

```
<td colspan="数值">单元格内容</td>
```

其中，colspan 属性的取值为数值型整数数据，代表几个单元格进行左右合并。

例如，在例 9.1 表格的基础上，将 A1 单元格和 B1 单元格合并成一个单元格。为第一行的第一个<td>标记增加 colspan="2"属性，并且将 B1 单元格的<td>标记删除。

【例 9.9】用 colspan 属性合并左右单元格(案例文件：ch09\9.9.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>单元格左右合并</title>
</head>
<body>
<table border="1">
  <tr>
    <td colspan="2">A1 B1</td>
    <td>C1</td>
  </tr>
  <tr>
    <td>A2</td>
    <td>B2</td>
    <td>C2</td>
  </tr>
  <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
  </tr>
  <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
  </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页，效果如图 9-10 所示。

从预览图中可以看到，A1 和 B1 单元格合并成一个单元格，C1 还在原来的位置上。



图 9-10　单元格左右合并



合并单元格以后，相应的单元格标记就应该减少。例如，A1 和 B1 合并后，B1 单元格的<td></td>标记就应该丢掉，否则单元格就会多出一个，并且后面的单元格依次向右位移。

2. 用 rowspan 属性合并上下单元格

上下单元格的合并需要为<td>标记增加 rowspan 属性，其语法格式如下：

```
<td rowspan="数值">单元格内容</td>
```

其中，rowspan 属性的取值为数值型整数数据，代表几个单元格进行上下合并。

例如，在例 9.1 表格的基础上，将 A1 单元格和 A2 单元格合并成一个单元格。为第一行的第一个<td>标记增加 rowspan="2"属性，并且将 A2 单元格的<td>标记删除。

【例 9.10】用 rowspan 属性合并上下单元格(案例文件：ch09\9.10.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>单元格上下合并</title>
</head>
<body>
<table border="1">
  <tr>
    <td rowspan="2">A1</td>
    <td>B1</td>
    <td>C1</td>
  </tr>
  <tr>
    <td>B2</td>
    <td>C2</td>
  </tr>
  <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
  </tr>
  <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
  </tr>
</table>
</body>
</html>
```



图 9-11 单元格上下合并

在 IE 11.0 中预览网页，效果如图 9-11 所示。

从预览图中可以看到，A1 单元格和 A2 单元格合并成了一个单元格。

通过上面对左右单元格合并和上下单元格合并的操作，读者会发现，合并单元格就是“丢掉”某些单元格。对于左右合并，就是以左侧为准，将右侧要合并的单元格“丢掉”；对于上下合并，就是以上方为准，将下方要合并的单元格“丢掉”。如果一个单元格既要向右合并，又要向下合并，该如何实现呢？

【例 9.11】单元格向右和向下合并(案例文件：ch09\9.11.html)。

```
<!DOCTYPE html>
```



```
<html>
<head>
<title>单元格上下左右合并</title>
</head>
<body>
<table border="1">
  <tr>
    <td colspan="2" rowspan="2">A1B1<br>A2B2</td>
    <td>C1</td>
  </tr>
  <tr>
    <td>C2</td>
  </tr>
  <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
  </tr>
  <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
  </tr>
</table>
</body>
</html>
```

图 9-12 在两个方向合并单元格

在 IE 11.0 中预览网页，效果如图 9-12 所示。

从上述代码可以看到，A1 单元格向右合并 B1 单元格，向下合并 A2 单元格，并且 A2 单元格向右合并 B2 单元格。

3. 使用 Dreamweaver CC 合并单元格

使用 HTML 创建表格非常麻烦。在 Dreamweaver CC 中，提供了表格的快捷操作，类似于在 Word 中编辑表格的操作。在 Dreamweaver CC 中创建表格，只需要选择“插入”→“表格”菜单命令，在弹出的对话框中指定表格的行数、列数、宽度和边框，即可在光标处创建一个空白表格。选择表格后，属性面板提供了表格的常用操作，如图 9-13 所示。



图 9-13 表格的属性面板

注意 表格属性面板中的操作，请结合前面讲述的 HTML 语言。对于按钮命令，可将鼠标悬停于按钮之上，数秒之后会出现命令提示。

关于表格的操作不再赘述，读者可自行操作，这里重点讲解如何使用 Dreamweaver CC 合并单元格。在 Dreamweaver CC 可视化操作中，提供了合并与拆分单元格两种操作。拆分单元格的操作，其实还是进行合并的操作。进行单元格合并和拆分时，应将光标置于单元格内，

如果选择了一个单元格，拆分命令有效，如图 9-14 所示。如果选择了两个或两个以上单元格，则合并命令有效。

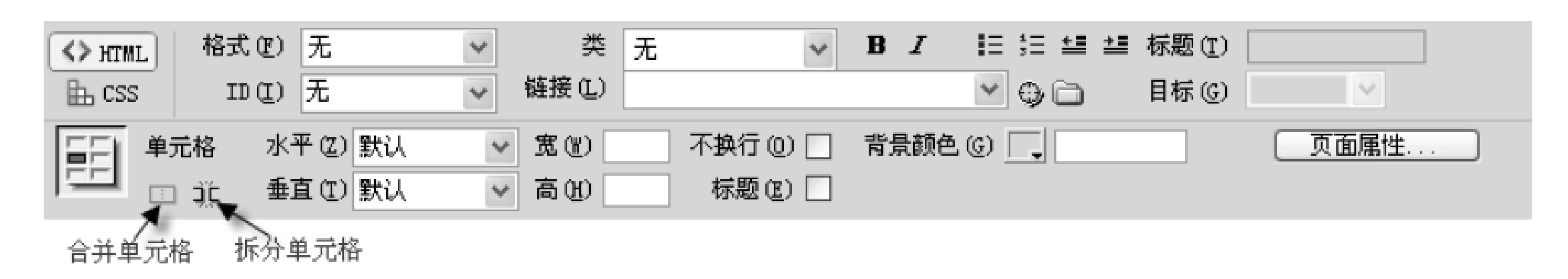


图 9-14 拆分单元格有效

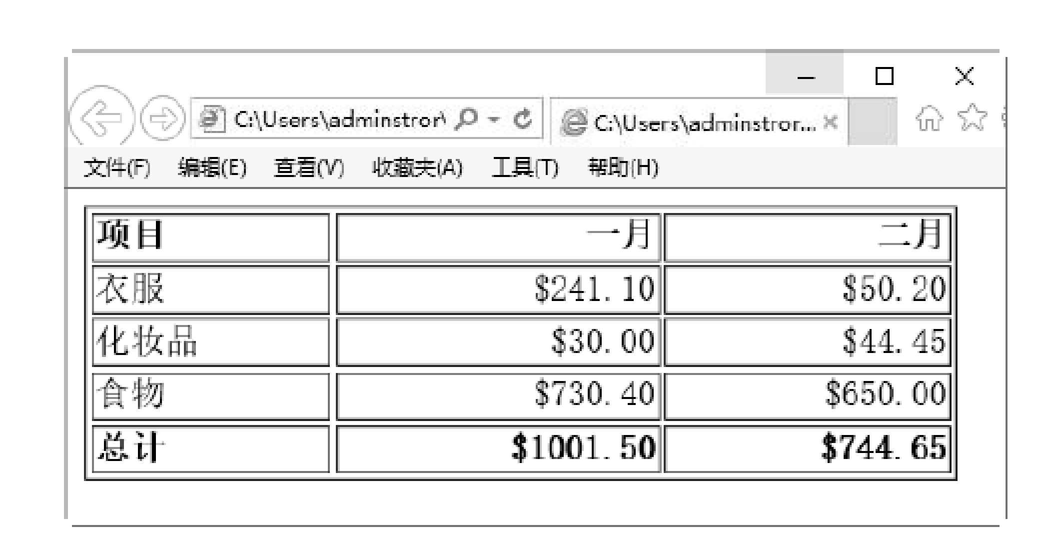
9.3.6 案例 9——排列单元格中的内容

使用 align 属性可以排列单元格中的内容，以便创建一个美观的表格。

【例 9.12】排列单元格中的内容(案例文件：ch09\9.12.html)。

```
<!DOCTYPE html>
<html>
<body>
<table width="400" border="1">
 <tr>
  <th align="left">项目</th>
  <th align="right">一月</th>
  <th align="right">二月</th>
 </tr>
 <tr>
  <td align="left">衣服</td>
  <td align="right">$241.10</td>
  <td align="right">$50.20</td>
 </tr>
 <tr>
  <td align="left">化妆品</td>
  <td align="right">$30.00</td>
  <td align="right">$44.45</td>
 </tr>
 <tr>
  <td align="left">食物</td>
  <td align="right">$730.40</td>
  <td align="right">$650.00</td>
 </tr>
 <tr>
  <th align="left">总计</th>
  <th align="right">$1001.50</th>
  <th align="right">$744.65</th>
 </tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页，效果如图 9-15 所示。



项目	一月	二月
衣服	$241.10	$50.20
化妆品	$30.00	$44.45
食物	$730.40	$650.00
总计	$1001.50	$744.65

图 9-15 排列单元格的内容

9.3.7 案例 10——设置单元格的行高与列宽

使用 cellpadding 来创建单元格内容与其边框之间的空白，从而调整表格的行高与列宽。

【例 9.13】设置单元格的行高与列宽(案例文件：ch09\9.13.html)。

```
<!DOCTYPE html>
<html>
<body>
<h4>调整前</h4>
<table border="1">
<tr>
  <td>1000</td>
  <td>2000</td>
</tr>
<tr>
  <td>2000</td>
  <td>3000</td>
</tr>
</table>
<h4>调整后</h4>
<table border="1" cellpadding="10">
<tr>
  <td>1000</td>
  <td>2000</td>
</tr>
<tr>
  <td>2000</td>
  <td>3000</td>
</tr>
</table>
</body>
</html>
```

在 IE 11.0 中预览网页，效果如图 9-16 所示。

图 9-16　使用 cellpadding 来调整表格的行高与列宽

9.4 案例 11——完整的表格标记

上面讲述了表格中最常用也是最基本的 3 个标记<table>、<tr>和<td>，使用它们可以构建出最简单的表格。为了让表格结构更清楚，以及配合后面学习的 CSS 样式，更方便地制作各种样式的表格，表格中还会出现表头、主体、脚注等。

按照表格结构，可以把表格的行分组，称为“行组”。不同的行组具有不同的意义。行组分为 3 类——“表头”“主体”和“脚注”。三者相应的 HTML 标记依次为<thead>、<tbody>和<tfoot>。

此外，在表格中还有两个标记：标记<caption>表示表格的标题；在一行中，除了<td>标记表示一个单元格以外，还可以使用<th>表示该单元格是这一行的“行头”。

【例 9.14】完整的表格标记(案例文件：ch09\9.14.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>完整表格标记</title>
<style>
tfoot{
    background-color:#FF3;
}
</style>
</head>
<body>
<table border="1">
  <caption>学生成绩单</caption>
  <thead>
    <tr>
      <th>姓名</th><th>性别</th><th>成绩</th>
    </tr>
  </thead>
   <tfoot>
    <tr>
      <td>平均分</td><td colspan="2">540</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>张三</td><td>男</td><td>560</td>
    </tr>
    <tr>
      <td>李四</td><td>男</td><td>520</td>
    </tr>
  </tbody>
</table>
</body>
</html>
```

从上述代码可以发现，使用 caption 表格定义了表格标题，<thead>、<tbody>和<tfoot>标记对表格进行了分组。在<thead>部分使用<th>标记代替<td>标记定义单元格，<th>标记定义的单元格内容默认加粗显示。网页的预览效果如图 9-17 所示。

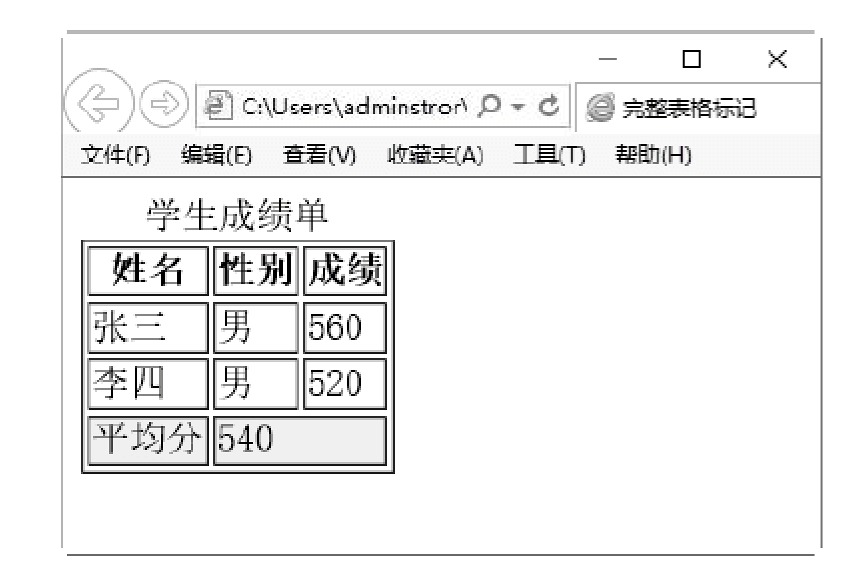
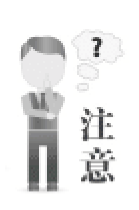


图 9-17　完整的表格结构



<caption>标记必须紧随<table>标记之后。

9.5　综合案例——制作计算机报价表

利用所学的表格知识，来制作如图 9-18 所示的计算机报价表。

具体操作步骤如下。

step 01 新建 HTML 5 文档，代码如下：



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>完整表格标记</title>
</head>
<body>
</body>
</html>
```

计算机报价单

型号	类型	价格	图片
宏碁 (Acer) AS4552-P362G32MNCC	笔记本	￥2799	
戴尔 (Dell) 14VR-188	笔记本	￥3499	
联想 (Lenovo) G470AH2310W42G500P7CW3(DB)-CN	笔记本	￥4149	
戴尔家用 (DELL) I560SR-656	台式	￥3599	
宏图奇眩(Hiteker) HS-5508-TF	台式	￥3399	
联想 (Lenovo) G470	笔记本	￥4299	

图 9-18　计算机报价表

step 02 在 HTML 文档的 body 部分增加表格及内容，代码如下：

```
<table>
  <caption>计算机报价单</caption>
   <tr>
     <th>型号</th>
     <th>类型</th>
     <th>价格</th>
     <th>图片</th>
   </tr>
   <tr>
     <td>宏碁(Acer) AS4552-P362G32MNCC</td>
     <td>笔记本</td>
     <td>￥2799</td>
     <td><img src="images/Acer.jpg" width="120" height="120"></td>
   </tr>
```

```
    <tr>
      <td>戴尔(Dell) 14VR-188</td><td>笔记本</td>
      <td>￥3499</td>
      <td><img src="images/Dell.jpg" width="120" height="120"></td>
    </tr>
     <tr>
      <td>联想(Lenovo) G470AH2310W42G500P7CW3(DB)-CN  </td>
      <td>笔记本</td>
      <td>￥4149</td>
      <td><img src="images/Lenovo.jpg" width="120" height="120"></td>
    </tr>
    <tr>
      <td>戴尔家用(DELL)  I560SR-656</td>
      <td>台式</td>
      <td>￥3599</td>
      <td><img src="images/DellT.jpg" width="120" height="120"></td>
    </tr>
    <tr>
      <td>宏图奇眩(Hiteker)  HS-5508-TF</td>
      <td>台式</td>
      <td>￥3399</td>
      <td><img src="images/Hiteker.jpg" width="120" height="120"></td>
    </tr>
    <tr>
      <td>联想(Lenovo) G470</td>
      <td>笔记本</td>
      <td>￥4299</td>
      <td><img src="images/LenovoG.jpg" width="120" height="120"></td>
    </tr>
</table>
```

利用 caption 标记制作表格的标题，用<th>代替<td>作为标题行单元格。可以将图片放在单元格内，即在<td>标记内使用<img>标记。

step 03 在 HTML 文档的 head 部分增加 CSS 样式，为表格增加边框及相应的修饰，代码如下：

```
<style>
table{
    /*表格增加线宽为 3 的橙色实线边框*/
    border:3px solid #F60;
}
caption{
    /*表格标题字号 36*/
    font-size:36px;
}
th,td{
    /*表格单元格(th、td)增加边线*/
    border:1px solid #F90;
}
</style>
```

step 04 保存网页后，即可查看最终效果。

9.6 跟我学上机——制作学生成绩表

本练习将结合前面学习的知识，创建一个学生成绩表。具体操作步骤如下。

step 01 分析需求。首先需要建立一个表格，所有行的颜色不单独设置，统一采用表格本身的背景色。然后根据 CSS 设置可以实现该效果，如图 9-19 所示。

step 02 创建 HTML 网页，实现 table 表格：

```
<!DOCTYPE html>
<html>
<head>
<title>学生成绩表</title>
</head>
<body>
<table border="0" cellpadding="0" cellspacing="1">
<caption>学生成绩表</caption>
   <tr>
     <th>姓名</th>
     <th>语文成绩</th>
   </tr>
   <tr class="hui">
     <td>王锋</td>
     <td>85</td>
   </tr>
   <tr>
     <td>李伟</td>
     <td>78</td>
   </tr>
   <tr class="hui">
     <td>张宇</td>
     <td>89</td>
   </tr>
   <tr>
     <td>苏石</td>
     <td>86</td>
   </tr>
   <tr class="hui">
     <td>马丽</td>
     <td>90</td>
   </tr>
   <tr>
     <td>张丽</td>
     <td>90</td>
   </tr>
   <tr class="hui">
     <td>冯尚</td>
     <td>85</td>
   </tr>
   <tr>
     <td >李旺</td>
     <td>75</td>
```

```
    </tr>
</table>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 9-20 所示，可以看到显示了一个表格，表格不带有边框，字体等都是默认显示。

学生成绩表

姓名	语文成绩
王锋	85
李伟	78
张宇	89
苏石	86
马丽	90
张丽	90
冯尚	85
李旺	75

图 9-19　变色表格

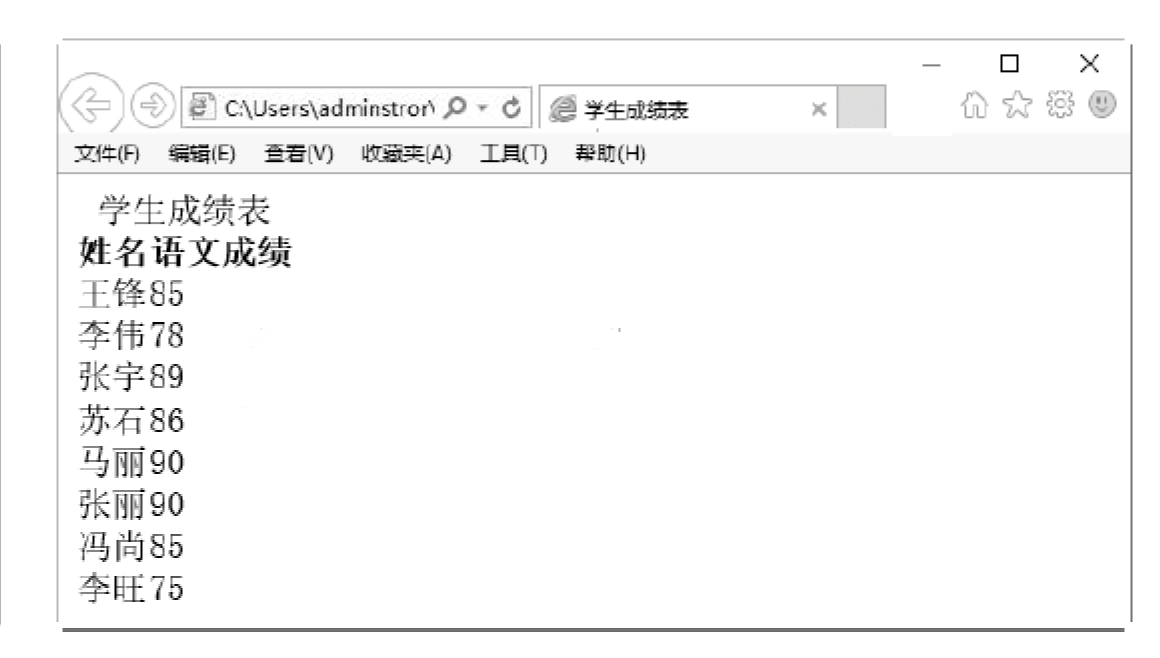


图 9-20　创建基本表格

step 03 添加 CSS 代码，修饰 table 表格和单元格：

```
<style type="text/css">
<!--
table {
    width: 600px;
    margin-top: 0px;
    margin-right: auto;
    margin-bottom: 0px;
    margin-left: auto;
    text-align: center;
    background-color: #000000;
    font-size: 9pt;
}
td {
    padding: 5px;
    background-color: #FFFFFF;
}
-->
</style>
```

在 IE 11.0 中浏览，效果如图 9-21 所示，可以看到显示了一个表格，表格带有边框，行内字体居中显示，但列标题背景色为黑色，其中字体不能够显示。

step 04 添加 CSS 代码，修饰标题：

```
caption{
    font-size: 36px;
    font-family: "黑体", "宋体";
    padding-bottom: 15px;
}
tr{
    font-size: 13px;
    background-color: #cad9ea;
```

```
    color: #000000;
}
th{
    padding: 5px;
}
.hui td {
    background-color: #f5fafe;
}
```

在上述代码中，使用了类选择器 hui 来定义每个 td 行所显示的背景色，此时需要在表格中每个奇数行都引入该类选择器。例如<tr class="hui">，从而设置奇数行的背景色。

在 IE 11.0 中浏览，效果如图 9-22 所示。可以看到，一个表格中列标题一行背景色显示为浅蓝色，并且表格中奇数行背景色为浅灰色，而偶数行背景色为默认的白色。

学生成绩表

王锋	85
李伟	78
张宇	89
苏石	86
马丽	90
张丽	90
冯尚	85
李旺	75

图 9-21　设置 table 样式

学生成绩表

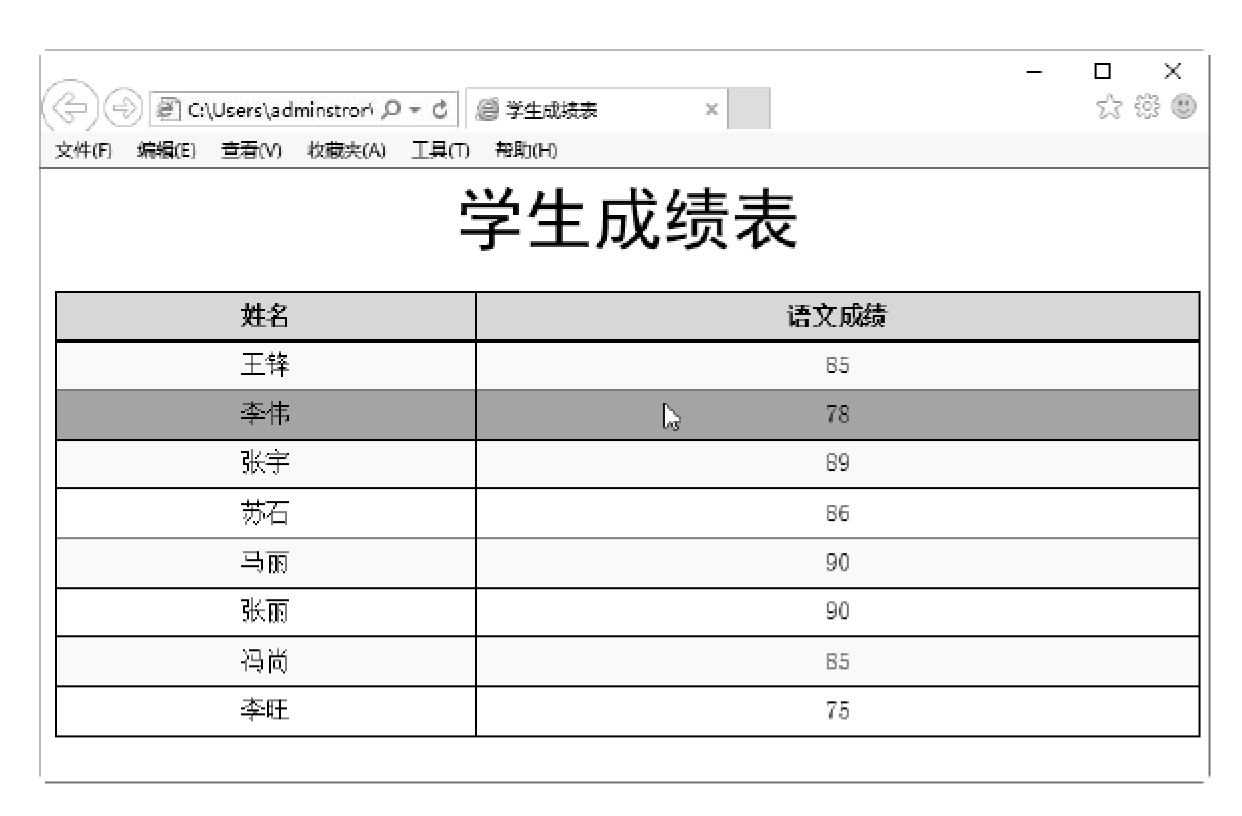
姓名	语文成绩
王锋	85
李伟	78
张宇	89
苏石	86
马丽	90
张丽	90
冯尚	85
李旺	75

图 9-22　设置奇数行背景色

step 05 添加 CSS 代码，实现鼠标悬浮变色：

```
tr:hover td {
    background-color: #FF9900;
}
```

在 IE 11.0 中浏览，效果如图 9-23 所示。可以看到，当鼠标放到不同行上面时，其背景会显示不同的颜色。



学生成绩表

姓名	语文成绩
王锋	85
李伟	78
张宇	89
苏石	86
马丽	90
张丽	90
冯尚	85
李旺	75

图 9-23　鼠标悬浮改变颜色

9.7 高手解惑

疑问 1：在 Dreamweaver CC 中如何选择多个单元格？

答：在 Dreamweaver CC 中选择单元格的操作类似于文字处理工具 Word，按住鼠标左键拖动鼠标，经过的单元格都会被选中。按住 Ctrl 键，单击某个单元格，该单元格将会被选中，这些单元格可以是连续的，也可以是不连续的。在需要选择区域的开头单元格中单击，按住 Shift 键，在区域的末尾单元格中单击，开头和结尾单元格组成的区域内的所有单元格将会被选中。

疑问 2：表格除了显示数据，还可以进行布局，为何不使用表格进行布局？

答：在互联网刚刚开始普及时，网页非常简单，形式也很单调，当时美国的 David Siegel 发明了使用表格布局，风靡全球。在表格布局的页面中，表格不但需要显示内容，还要控制页面的外观及显示位置，导致页面代码过多，结构与内容无法分离，这样就给网站的后期维护和其他方面带来了麻烦。

疑问 3：使用<thead>、<tbody>和<tfoot>标记对行进行分组的意义何在？

答：在 HTML 文档中增加<thead>、<tbody>和<tfoot>标记，虽然从外观上不能看出任何变化，但是它们能使文档的结构更加清晰。使用<thead>、<tbody>和<tfoot>标记除了使文档更加清晰外，还有一个更重要的意义，就是方便使用 CSS 样式对表格的各个部分进行修饰，从而制作出更炫的表格。



第 10 章 HTML 5 中的音频和视频

目前，在网页上没有关于音频和视频的标准，多数音频和视频都是通过插件来播放的。为此，HTML 5 新增了音频和视频的标签。本章将讲述音频和视频的基本概念、常用属性和浏览器的支持情况。



10.1　audio 标签概述

目前，大多数音频是通过插件来播放音频文件的，如常见的播放插件为 Flash。这就是为什么用户在用浏览器播放音乐时，常常需要安装 Flash 插件的原因。但是，并不是所有的浏览器都拥有同样的插件。为此，与 HTML 4 相比，HTML 5 新增了 audio 标签，规定了一种包含音频的标准方法。

10.1.1　案例 1——认识 audio 标签

audio 标签主要是定义播放声音文件或者音频流的标准。它支持 3 种音频格式，分别为 Ogg、MP3 和 WAV。

如果需要在 HTML 5 网页中播放音频，输入的基本格式如下：

```
<audio src="song.mp3" controls="controls"></audio>
```



其中，src 属性规定要播放的音频的地址，controls 属性是供添加播放、暂停和音量控件的属性。

另外，在<audio>和</audio>之间插入的内容是供不支持 audio 元素的浏览器显示的。

【例 10.1】认识 audio 标签(案例文件：ch10\10.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>audio</title>
<head>
<body>
<audio src="song.mp3" controls="controls">
   您的浏览器不支持 audio 标签！
</audio>
</body>
</html>
```

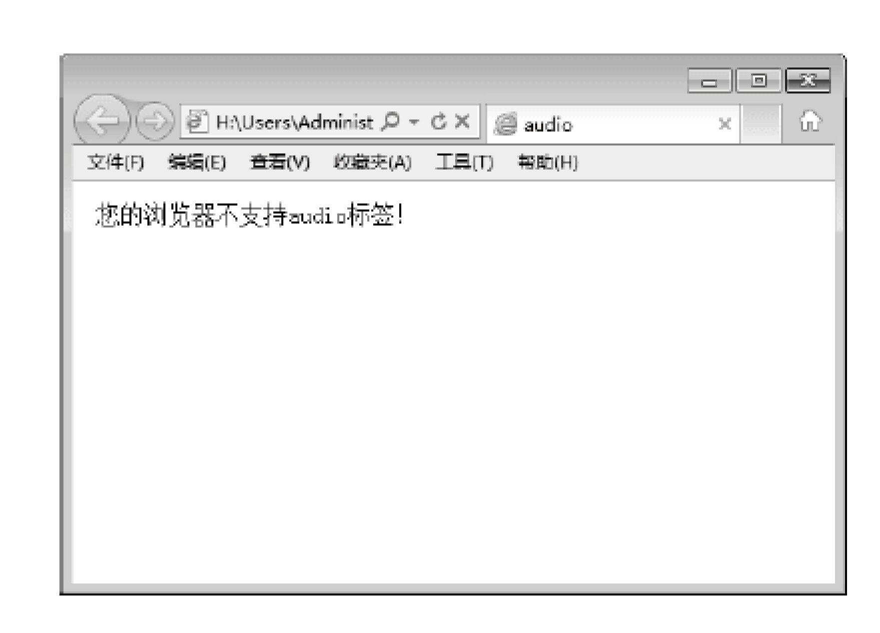


图 10-1　不支持 audio 标签的效果

如果用户的浏览器是 IE 11.0 以前的版本，浏览效果如图 10-1 所示，可见 IE 11.0 以前的浏览器版本不支持 audio 标签。

在 IE 11.0 中浏览，效果如图 10-2 所示，可以看到加载的音频控制条并听到声音，此时用户还可以控制音量的大小。



图 10-2　支持 audio 标签的效果



10.1.2 audio 标签的属性

audio 标签的常见属性和含义如表 10-1 所示。

表 10-1 audio 标签的常见属性

属 性	值	描 述
autoplay	autoplay(自动播放)	如果出现该属性，则音频在就绪后马上播放
controls	controls(控制)	如果出现该属性，则向用户显示控件，比如播放按钮
loop	loop(循环)	如果出现该属性，则每当音频结束时重新开始播放
preload	preload(加载)	如果出现该属性，则音频在页面加载时进行加载，并预备播放。如果使用 autoplay，则忽略该属性
src	url(地址)	要播放的音频的 URL 地址

另外，audio 标签可以通过 source 属性添加多个音频文件，具体格式如下：

```
<audio controls="controls">
   <source src="123.ogg" type="audio/ogg">
   <source src="123.mp3" type="audio/mpeg">
</audio>
```

10.1.3 浏览器对 audio 标签的支持情况

目前，不同的浏览器对 audio 标签的支持情况也不同。表 10-2 中列出了应用最为广泛的浏览器对 audio 标签的支持情况。

表 10-2 浏览器对 audio 标签的支持情况

音频格式	浏 览 器				
	Firefox 3.5 及更高版本	IE 11.0 及更高版本	Opera 10.5 及更高版本	Chrome 3.0 及更高版本	Safari 3.0 及更高版本
Ogg Vorbis	支持		支持	支持	
MP3		支持		支持	支持
WAV	支持		支持		支持

10.2 在网页中添加音频文件

当在网页中添加音频文件时，用户可以根据自己的需要，添加不同类型的音频文件，如添加自动播放的音频文件、添加带有控件的音频文件、添加循环播放的音频文件和添加预播放的音频文件等。

10.2.1 案例 2——添加自动播放的音频文件

autoplay 属性规定一旦音频就绪，马上就开始播放。如果设置了该属性，音频将自动播

放。下面就是在网页中添加自动播放音频文件的相关代码。

【例 10.2】添加自动播放的音频文件(案例文件：ch10\10.2.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<audio controls="controls" autoplay="autoplay">
    <source src="song.mp3">
</audio>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-3 所示。可以看到，网页中加载了音频播放控制条，并开始自动播放加载的音频文件。

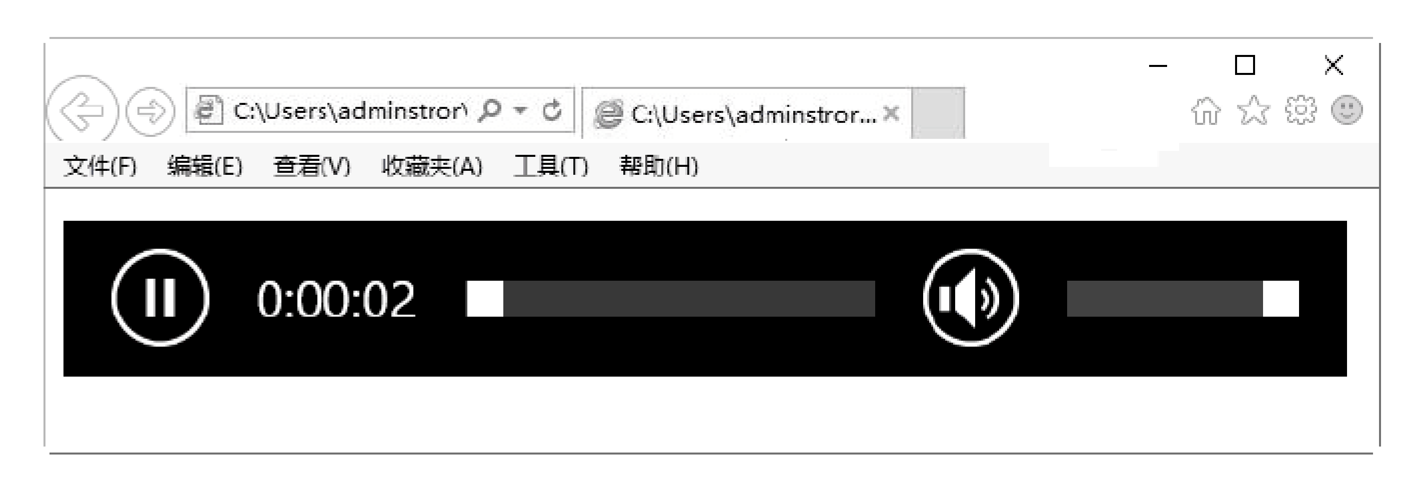


图 10-3　添加自动播放的音频文件

10.2.2　案例 3——添加带有控件的音频文件

controls 属性规定浏览器应该为音频提供播放控件。如果设置了该属性，则规定不存在作者设置的脚本控件。其中浏览器控件应该包括播放、暂停、定位、音量、全屏切换等。

【例 10.3】添加带有控件的音频文件(案例文件：ch10\10.3.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<audio controls="controls">
    <source src="song.mp3">
</audio>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-4 所示。可以看到，网页中加载了音频播放控制条，这里只有单击其中的“播放”按钮，才可以播放加载的音频文件。



图 10-4　添加带有控件的音频文件



10.2.3 案例 4——添加循环播放的音频文件

loop 属性规定当音频结束后将重新开始播放。如果设置该属性，则音频将循环播放。

【例 10.4】添加循环播放的音频文件(案例文件：ch10\10.4.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<audio controls="controls" loop="loop">
   <source src="song.mp3">
</audio>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-5 所示。可以看到，网页中加载了音频播放控制条，单击“播放”按钮，开始播放加载的音频文件。播放完毕后，音频文件将重新开始播放。



图 10-5 添加循环播放的音频文件

10.2.4 案例 5——添加预播放的音频文件

preload 属性规定是否在页面加载后载入音频。如果设置了 autoplay 属性，则忽略该属性。preload 属性的值可能有 3 种，分别如下。

(1) auto：当页面加载后载入整个音频。

(2) meta：当页面加载后只载入元数据。

(3) none：当页面加载后不载入音频。

【例 10.5】添加预播放的音频文件(案例文件：ch10\10.5.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<audio controls="controls" preload="auto">
   <source src="song.mp3">
</audio>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-6 所示。可以看到，网页中加载了音频播放控制条。

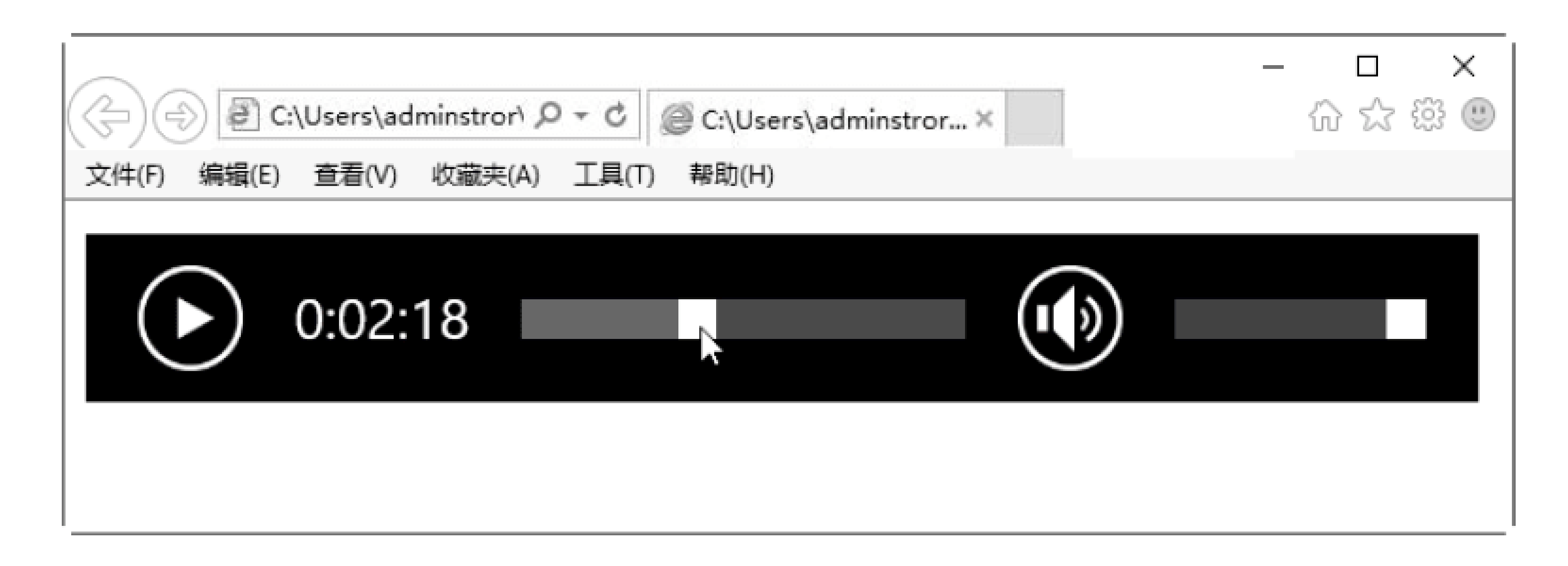


图 10-6　添加预播放的音频文件

10.3　video 标签概述

与音频文件播放方式一样，大多数视频文件在网页上也是通过插件来播放的，如常见的播放插件为 Flash。由于不是所有的浏览器都拥有同样的插件，所以就需要一种统一的包含视频的标准方法。为此，与 HTML 4 相比，HTML 5 新增了 video 标签。

10.3.1　案例 6——认识 video 标签

video 标签主要是定义播放视频文件或者视频流的标准。它支持 3 种视频格式，分别为 Ogg、WebM 和 MPEG 4。

如果需要在 HTML 5 网页中播放视频，输入的基本格式如下：

```
<video src="123.mp4" controls="controls">...</video>
```

其中，在<video>与</video>之间插入的内容是供不支持 video 元素的浏览器显示的。

【例 10.6】 video 标签的使用(案例文件：ch10\10.6.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>video</title>
<head>
<body>
<video src="movie.mp4" controls="controls">
    您的浏览器不支持 video 标签!
</video>
</body>
</html>
```

如果用户的浏览器是 IE 11.0 以前的版本，浏览效果如图 10-7 所示，可见 IE 11.0 以前版本的浏览器不支持 video 标签。

在 IE 11.0 中浏览，效果如图 10-8 所示，可以看到加载的视频控制条界面。单击“播放”按钮，即可查看视频的内容，同时用户还可以调整音量的大小。




图 10-7　不支持 video 标签的效果

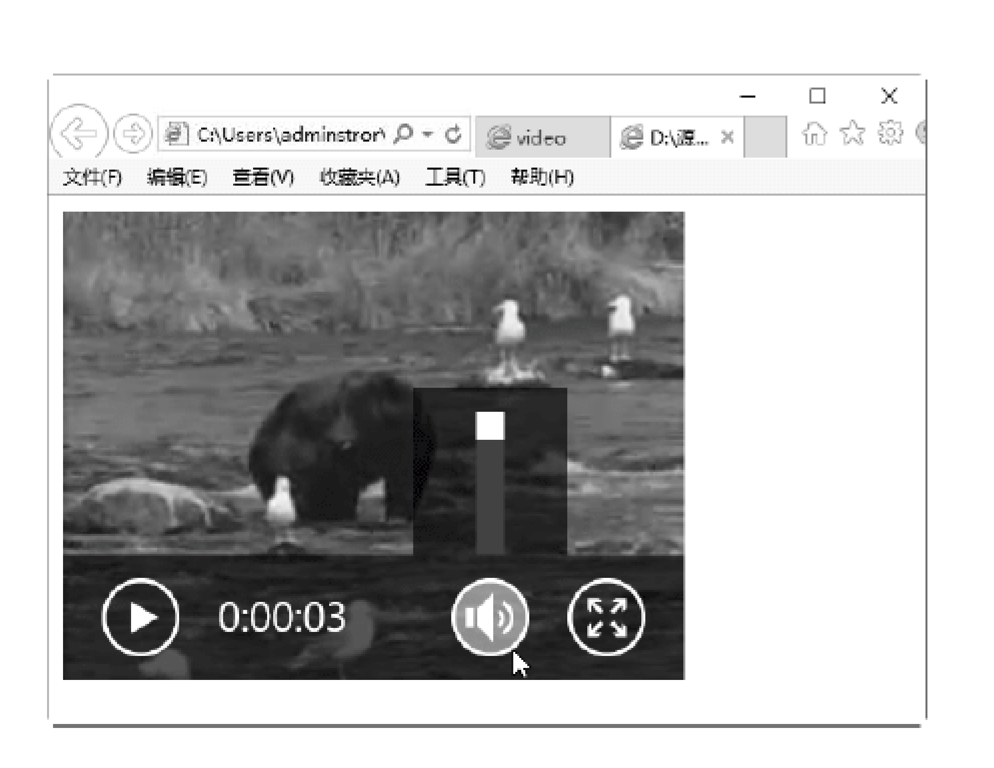


图 10-8　支持 video 标签的效果

10.3.2　video 标签的属性

video 标签的常见属性及其含义如表 10-3 所示。

表 10-3　video 标签的常见属性

属　性	值	描　述
autoplay	autoplay	视频就绪后马上播放
controls	controls	向用户显示控件，比如“播放”按钮
loop	loop	每当视频结束时重新开始播放
preload	preload	视频在页面加载时进行加载，并预备播放。如果使用 autoplay，则忽略该属性
src	url	要播放的视频的 URL
width	宽度值	设置视频播放器的宽度
height	高度值	设置视频播放器的高度
poster	url	当视频未响应或缓冲不足时，该属性值链接到一个图像。该图像将以一定比例被显示出来

由表 10-3 可知，用户可以自定义视频文件显示的大小。例如，如果想让视频以 320 像素×240 像素大小显示，可以加入 width 和 height 属性。具体格式如下：

```
<video width="320" height="240" controls src="movie.mp4"></video>
```

另外，video 标签可以通过 source 属性添加多个视频文件，具体格式如下：

```
<video controls="controls">
   <source src="123.ogg" type="video/ogg">
   <source src="123.mp4" type="video/mp4">
</video>
```

10.3.3　浏览器对 video 标签的支持情况

目前，不同的浏览器对 video 标签的支持情况也不同。表 10-4 中列出了应用最为广泛的浏览器对 video 标签的支持情况。

表 10-4　浏览器对 video 标签的支持情况

视频格式	浏 览 器				
	Firefox 4.0 及更高版本	IE 11.0 及更高版本	Opera 10.6 及更高版本	Chrome 10.0 及更高版本	Safari 3.0 及更高版本
Ogg	支持		支持	支持	
MPEG 4		支持		支持	支持
WebM	支持		支持	支持	

10.4　在网页中添加视频文件

当在网页中添加视频文件时，用户可以根据自己的需要添加不同类型的视频文件，如添加自动播放的视频文件、添加带有控件的视频文件、添加循环播放的视频文件等，另外，还可以设置视频文件的高度和宽度。

10.4.1　案例 7——添加自动播放的视频文件

autoplay 属性规定一旦视频就绪马上开始播放。如果设置了该属性，视频将自动播放。

【例 10.7】添加自动播放的视频文件(案例文件：ch10\10.7.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<video controls="controls" autoplay="autoplay">
   <source src="movie.mp4">
</video>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-9 所示。可以看到，网页中加载了视频播放控件，并开始自动播放加载的视频文件。



图 10-9　添加自动播放的视频文件

10.4.2 案例 8——添加带有控件的视频文件

controls 属性规定浏览器应该为视频提供播放控件。如果设置了该属性，则规定不存在设置的脚本控件。其中浏览器控件应该包括播放、暂停、定位、音量、全屏切换等。

【例 10.8】添加带有控件的视频文件(案例文件：ch06\10.8.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<video controls="controls" controls="controls">
    <source src="movie.mp4">
</video>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-10 所示。可以看到，网页中加载了视频播放控件，这里只有单击其中的“播放”按钮，才可以播放加载的视频文件。



图 10-10 添加带有控件的视频文件

10.4.3 案例 9——添加循环播放的视频文件

loop 属性规定当视频结束后将重新开始播放。如果设置该属性，则视频将循环播放。

【例 10.9】添加循环播放的视频文件(案例文件：ch10\10.9.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<video controls="controls" loop="loop">
    <source src="movie.mp4">
</video>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-11 所示。可以看到，网页中加载了视频播放控件，单击其中的“播放”按钮，开始播放加载的视频文件。播放完毕后，视频文件将重新开始播放。



图 10-11　添加循环播放的视频文件

10.5　综合案例——设置视频文件的高度与宽度

使用 width 和 height 属性可以设置视频文件的显示宽度与高度，单位是像素。

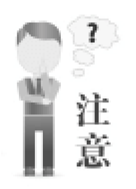


规定视频的高度和宽度是一个好习惯。如果设置这些属性，在页面加载时会为视频预留出空间。如果没有设置这些属性，那么浏览器就无法预先确定视频的尺寸，这样就无法为视频保留合适的空间。结果是，在页面加载的过程中，其布局也会产生变化。

【例 10.10】设置视频文件的高度与宽度(案例文件：ch10\10.10.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<video width="200" height="160" controls="controls">
   <source src="movie.mp4">
</video>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-12 所示。可以看到，网页中加载了视频播放控件，视频的显示大小为 200 像素×160 像素。



切勿通过 height 和 width 属性来缩放视频。通过 height 和 width 属性来缩小视频，用户仍会下载原始的视频(即使在页面上它看起来较小)。正确的方法是在网页上使用该视频前，用软件对视频进行压缩。

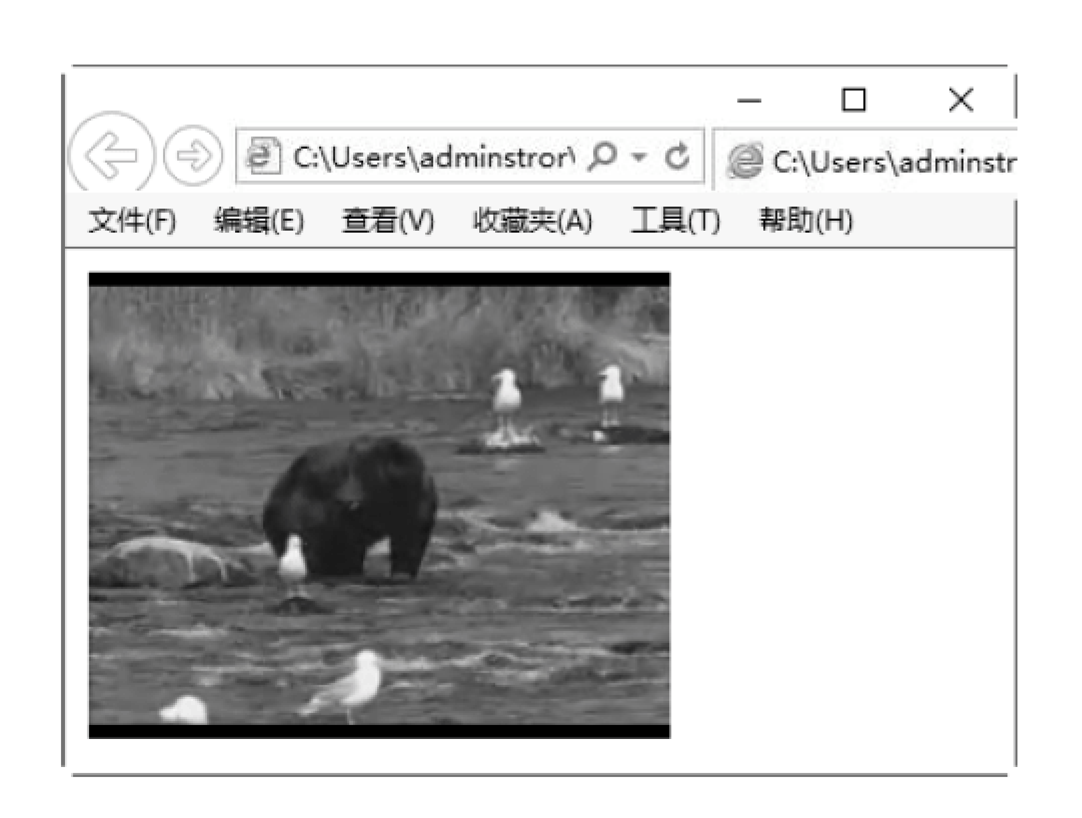


图 10-12　设置视频文件的高度与宽度

10.6　跟我学上机——添加预播放的视频文件

preload 属性规定是否在页面加载后载入视频。如果设置了 autoplay 属性，则忽略该属性。preload 属性的值可能有 3 种，分别说明如下。

(1)　auto：当页面加载后载入整个视频。

(2)　meta：当页面加载后只载入元数据。

(3)　none：当页面加载后不载入视频。

【例 10.11】添加预播放的视频文件(案例文件：ch10\10.11.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<video controls="controls" preload="auto">
   <source src="movie.mp4">
</video>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 10-13 所示。可以看到，网页中加载了视频播放控件。



图 10-13　添加预播放的视频文件

10.7 高手解惑

疑问 1：在 HTML 5 网页中添加所支持格式的视频，不能在 Firefox 53.0 浏览器中正常播放，为什么？

答：目前，HTML 5 的 video 标签对视频的支持，不仅有视频格式的限制，还有对解码器的限制。具体规定如下。

- Ogg 格式的文件需要 Thedora 视频编码和 Vorbis 音频编码。
- MPEG 4 格式的文件需要 H.264 视频编码和 AAC 音频编码。
- WebM 格式的文件需要 VP8 视频编码和 Vorbis 音频编码。

疑问 2：在 HTML 5 网页中添加 MP4 格式的视频文件，为什么在不同的浏览器中视频控件显示的外观不同？

答：在 HTML 5 中规定用 controls 属性来控制视频文件的播放、暂停、停止和调节音量的操作。controls 是一个布尔属性，一旦添加了此属性，等于告诉浏览器需要显示播放控件并允许用户进行操作。

因为每一个浏览器负责解释内置视频控件的外观，所以在不同的浏览器中，将会显示不同的视频控件外观。



第 11 章 使用 HTML 5 绘制图形

HTML 5 呈现了很多新特性，其中一个最值得提及的特性就是 HTML canvas。它可以对 2D 图形或位图进行动态、脚本的渲染。使用 canvas 可以绘制一个矩形区域，然后使用 JavaScript 可以控制其每一个像素。例如，可以用它来画图、合成图像，或制作简单的动画。本章介绍如何使用 HTML 5 绘制图形。



11.1　添加 canvas 的步骤

canvas 标签是一个矩形区域，它包含 width 和 height 两个属性，分别表示矩形区域的宽度和高度。这两个属性都是可选的，并且都可以通过 CSS 来定义，其默认值是 300px 和 150px。

canvas 在网页中的常用形式如下：

```
<canvas id="myCanvas" width="300" height="200"
  style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>
```

在上述示例代码中，id 表示画布对象名称，width 和 height 分别表示宽度和高度。最初的画布是不可见的，此处为了观察这个矩形区域，这里使用 CSS 样式，即 style 标记。style 表示画布的样式。如果浏览器不支持画布标记，会显示画布中间的提示信息。

画布 canvas 本身不具有绘制图形的功能，它只是一个容器。如果读者对于 Java 语言非常了解，就会发现 HTML 5 的画布和 Java 中的 Panel 面板非常相似，都可以在容器中绘制图形。既然 canvas 画布元素放好了，就可以使用脚本语言 JavaScript 在网页上绘制图形了。

使用 canvas 结合 JavaScript 绘制图形，一般情况下需要下面几个步骤。

step 01 JavaScript 使用 id 来寻找 canvas 元素，即获取当前画布对象：

```
var c = document.getElementById("myCanvas");
```

step 02 创建 context 对象：

```
var cxt = c.getContext("2d");
```

getContext 方法返回一个指定 contextId 的上下文对象。如果指定的 id 不被支持，则返回 null，当前唯一被强制必须支持的是 2d，也许在将来会有 3d。注意，指定的 id 是大小写敏感的。对象 cxt 建立之后，就可以拥有多种绘制路径、矩形、圆形、字符及添加图像的方法。

step 03 绘制图形：

```
cxt.fillStyle = "#FF0000";
cxt.fillRect(0,0,150,75);
```

fillStyle 方法将其染成红色，fillRect 方法规定了形状、位置和尺寸。这两行代码绘制一个红色的矩形。

11.2　绘制基本形状

画布 canvas 结合 JavaScript 可以绘制简单的矩形，还可以绘制一些其他的常见图形，如直线、圆等。

11.2.1 案例 1——绘制矩形

用 canvas 和 JavaScript 绘制矩形时，涉及一个或多个方法，这些方法如表 11-1 所示。

表 11-1 绘制矩形的方法

方 法	功 能
fillRect	绘制一个矩形，这个矩形区域没有边框，只有填充色。这个方法有 4 个参数，前两个表示左上角的坐标位置，第 3 个参数为长度，第 4 个参数为高度
strokeRect	绘制一个带边框的矩形。该方法的 4 个参数的解释同上
clearRect	清除一个矩形区域，被清除的区域将没有任何线条。该方法的 4 个参数的解释同上

【例 11.1】绘制矩形(案例文件：ch11\11.1.html)。

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="300" height="200"
  style="border:1px solid blue">
    Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c = document.getElementById("myCanvas");
var cxt = c.getContext("2d");
cxt.fillStyle = "rgb(0,0,200)";
cxt.fillRect(10,20,100,100);
</script>
</body>
</html>
```

在上述代码中，定义了一个画布对象，其 id 名称为 myCanvas，高度和宽度分别为 200 和 300 像素，并定义了画布边框的显示样式。代码中首先获取画布对象，然后使用 getContext 获取当前 2d 的上下文对象，并使用 fillRect 绘制一个矩形。其中涉及一个 fillStyle 属性，fillStyle 用于设定填充的颜色、透明度等，如果设置为 rgb(200,0,0)，则表示一个不透明颜色；如果设置为 rgba(0,0,200,0.5)，则表示一个透明度为 50%的颜色。

在 IE 11.0 中浏览，效果如图 11-1 所示。可以看到，在一个蓝色边框内显示了一个蓝色矩形。

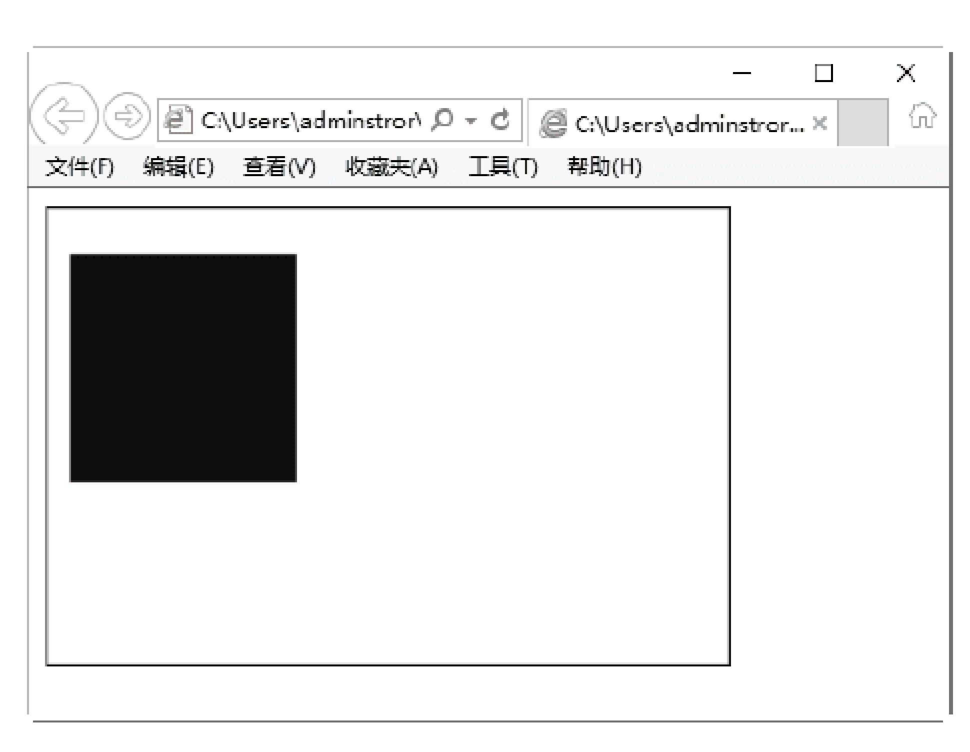


图 11-1 绘制矩形

11.2.2　案例 2——绘制圆形

在画布中绘制圆形，可能要涉及下面几个方法，如表 11-2 所示。

表 11-2　绘制圆形的方法

方　法	功　能
beginPath()	开始绘制路径
arc(x,y,radius,startAngle, endAngle,anticlockwise)	x 和 y 定义的是圆的原点；radius 是圆的半径；startAngle 和 endAngle 是弧度，不是度数；anticlockwise 用来定义画圆的方向，值是 true 或 false
closePath()	结束路径的绘制
fill()	进行填充
stroke()	该方法设置边框

路径是绘制自定义图形的好方法。在 canvas 中，通过 beginPath()方法开始绘制路径，这个时候，就可以绘制直线、曲线等。绘制完成后，调用 fill()和 stroke()方法完成填充和边框设置，通过 closePath()方法结束路径的绘制。

【例 11.2】绘制圆形(案例文件：ch11\11.2.html)。

```
<!DOCTYPE html>
<html><body>
<canvas id="myCanvas" width="200" height="200"
  style="border:1px solid blue">
    Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c = document.getElementById("myCanvas");
var cxt = c.getContext("2d");
cxt.fillStyle = "#FFaa00";
cxt.beginPath();
cxt.arc(70,18,15,0,Math.PI*2,true);
cxt.closePath();
cxt.fill();
</script>
</body></html>
```

在上面的 JavaScript 代码中，使用 beginPath()方法开启一个路径，然后绘制一个圆形，最后关闭这个路径并填充。在 IE 11.0 中浏览，效果如图 11-2 所示。

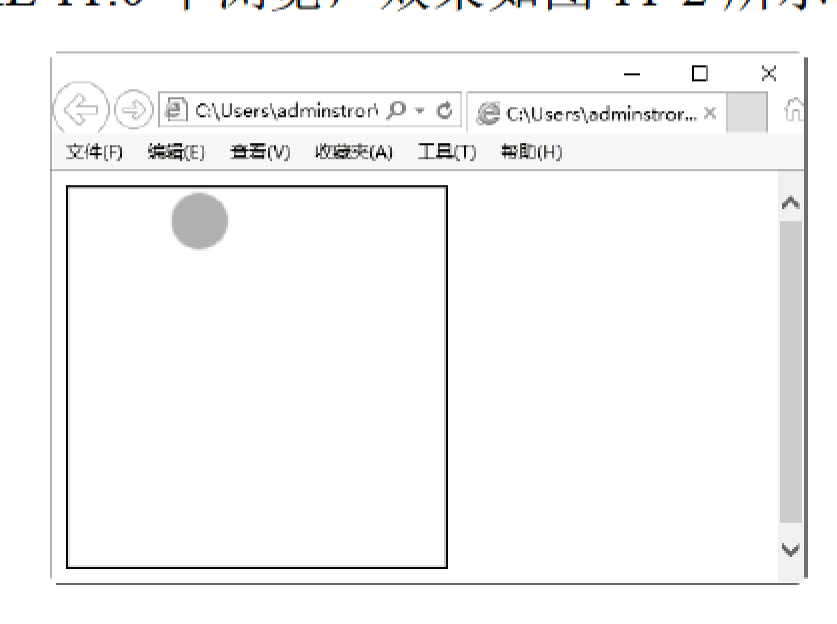


图 11-2　绘制圆形



11.2.3 案例 3——使用 moveTo 与 lineTo 绘制直线

绘制直线常用的方法是 moveTo 和 lineTo，其含义如表 11-3 所示。

表 11-3 绘制直线的方法

方法或属性	功 能
moveTo(x,y)	不绘制，只是将当前位置移动到新目标坐标(x,y)，并作为线条的开始点
lineTo(x,y)	绘制线条到指定的目标坐标(x,y)，并且在两个坐标之间画一条直线。不管调用它们哪一个，都不会真正画出图形，因为还没有调用 stroke 和 fill 函数。当前，只是在定义路径的位置，以便后面绘制时使用
strokeStyle	指定线条的颜色
lineWidth	设置线条的粗细

【例 11.3】使用 moveTo 与 lineTo 绘制直线(案例文件：ch11\11.3.html)。

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="200"
  style="border:1px solid blue">
   Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c = document.getElementById("myCanvas");
var cxt = c.getContext("2d");
cxt.beginPath();
cxt.strokeStyle = "rgb(0,182,0)";
cxt.moveTo(10,10);
cxt.lineTo(150,50);
cxt.lineTo(10,50);
cxt.lineWidth = 14;
cxt.stroke();
cxt.closePath();
</script>
</body>
</html>
```

在上述代码中，使用 moveTo()方法定义一个坐标位置为(10,10)，然后以此坐标位置为起点，绘制了两条不同的直线，并用 lineWidth 设置了直线的宽度，用 strokeStyle 设置了直线的颜色，用 lineTo()设置了两条不同直线的结束位置。

在 IE 11.0 中浏览，效果如图 11-3 所示。可以看到，网页中绘制了两条直线，这两条直线在某一点交叉。

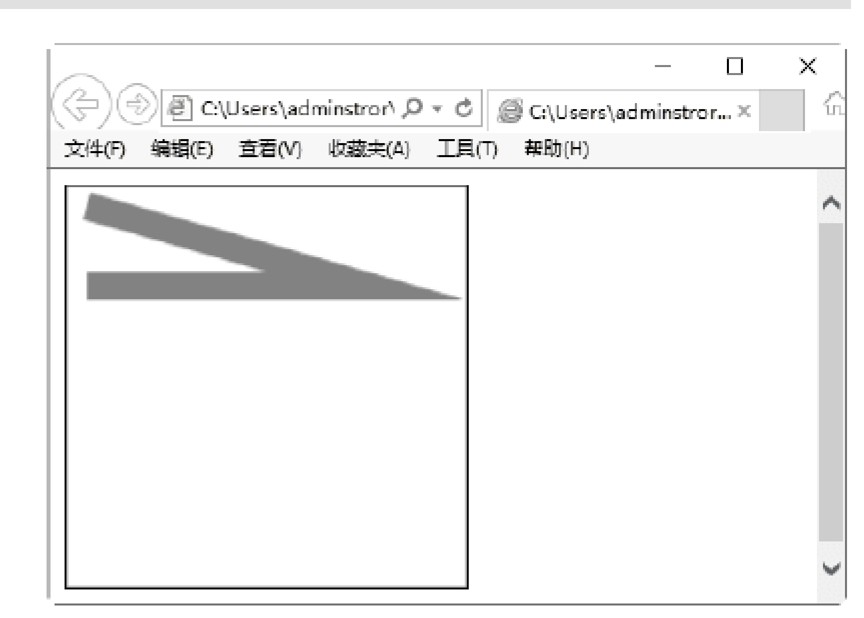


图 11-3 绘制直线

11.2.4　案例 4——使用 bezierCurveTo 绘制贝济埃曲线

在数学的数值分析领域中，贝济埃(Bézier)曲线是电脑图形学中非常重要的参数曲线。更高维度的广泛化贝济埃曲线就称作贝济埃曲面，其中贝济埃三角是一种特殊的实例。

bezierCurveTo()表示为一个画布的当前子路径添加一条三次贝济埃曲线。这条曲线的开始点是画布的当前点，而结束点是(x, y)。两条贝济埃曲线的控制点(cpX1, cpY1)和(cpX2, cpY2)定义了曲线的形状。当这个方法返回的时候，当前的位置为(x, y)。

方法 bezierCurveTo 的语法格式如下：

```
bezierCurveTo(cpX1, cpY1, cpX2, cpY2, x, y)
```

其参数的含义如表 11-4 所示。

表 11-4　绘制贝济埃曲线的参数

参　数	描　述
cpX1, cpY1	与曲线的开始点(当前位置)相关联的控制点的坐标
cpX2, cpY2	与曲线的结束点相关联的控制点的坐标
x, y	曲线的结束点的坐标

【例 11.4】使用 bezierCurveTo 绘制贝济埃曲线(案例文件：ch11\11.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>贝济埃曲线</title>
<script>
function draw(id)
{
    var canvas = document.getElementById(id);
    if(canvas==null)
        return false;
    var context = canvas.getContext('2d');
    context.fillStyle = "#eeeeff";
    context.fillRect(0,0,400,300);
    var n = 0;
    var dx = 150;
    var dy = 150;
    var s = 100;
    context.beginPath();
    context.globalCompositeOperation = 'and';
    context.fillStyle = 'rgb(100,255,100)';
    context.strokeStyle = 'rgb(0,0,100)';
    var x = Math.sin(0);
    var y = Math.cos(0);
    var dig = Math.PI/15*11;
    for(var i=0; i<30; i++)
    {
        var x = Math.sin(i*dig);
        var y = Math.cos(i*dig);
```

```
        context.bezierCurveTo(
          dx+x*s,dy+y*s-100,dx+x*s+100,dy+y*s,dx+x*s,dy+y*s);
    }
    context.closePath();
    context.fill();
    context.stroke();
}
</script>
</head>
<body onload="draw('canvas');">
<h1>绘制元素</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在上面的 draw()函数代码中，首先使用 fillRect(0,0,400,300)语句绘制了一个矩形，其大小与画布相同，填充颜色为浅青色。然后定义了几个变量，用于设定曲线的坐标位置，在 for 循环中使用 bezierCurveTo 绘制贝济埃曲线。在 IE 11.0 中浏览，效果如图 11-4 所示。可以看到，网页中显示了一个贝济埃曲线。

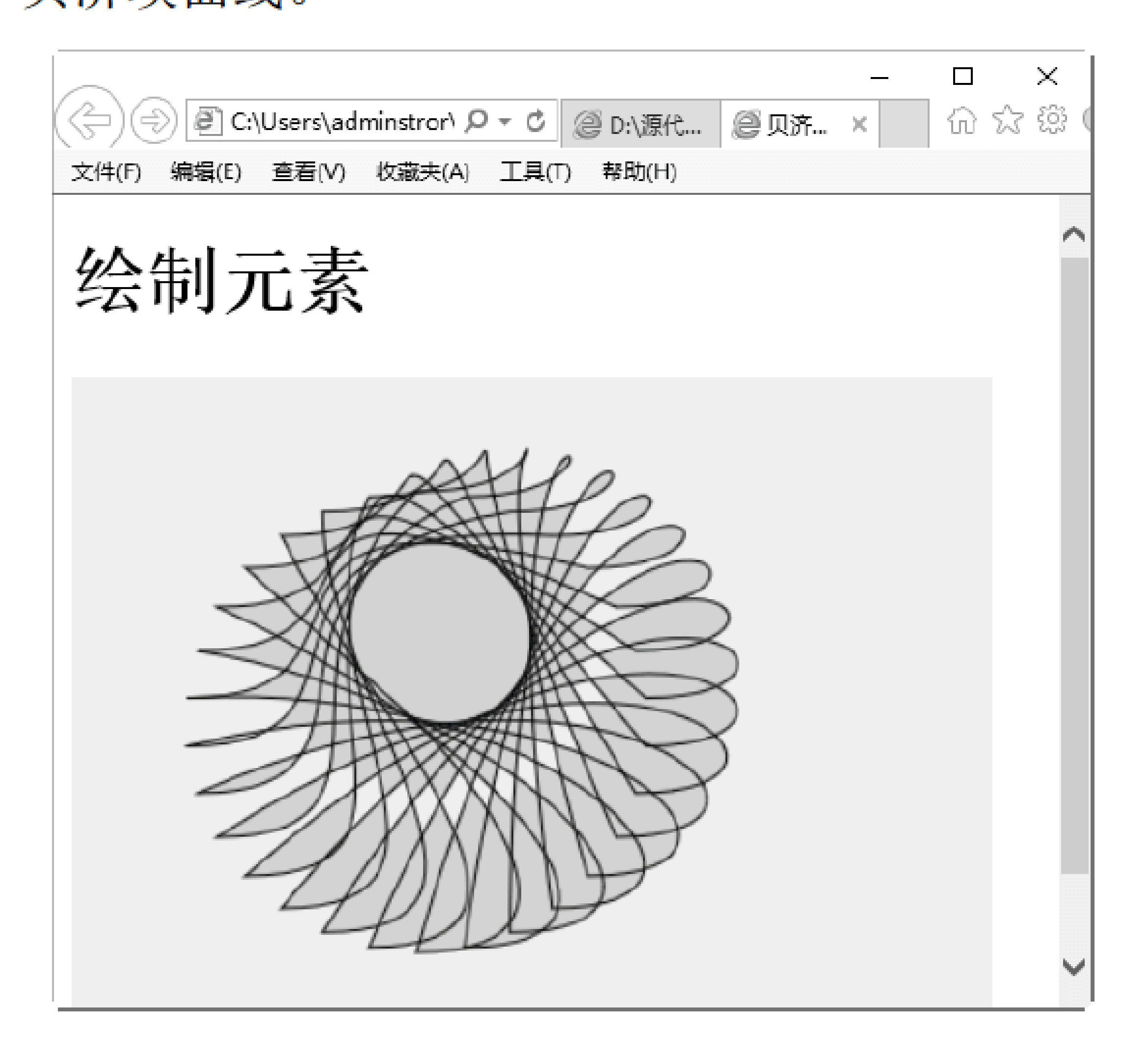


图 11-4　贝济埃曲线

11.3　绘制渐变图形

渐变是两种或更多颜色的平滑过渡，是指在颜色集上使用逐步抽样算法，并将结果应用于描边样式和填充样式中。canvas 的绘图上下文支持两种类型的渐变：线性渐变和放射性渐变，其中，放射性渐变也称为径向渐变。

11.3.1　案例 5——绘制线性渐变

创建一个简单的渐变非常容易。使用渐变需要如下 3 个步骤。

step 01 创建渐变对象：

```
var gradient = cxt.createLinearGradient(0,0,0,canvas.height);
```

step 02 为渐变对象设置颜色，指明过渡方式：

```
gradient.addColorStop(0,'#fff');
gradient.addColorStop(1,'#000');
```

step 03 在 context 上为填充样式或者描边样式设置渐变：

```
cxt.fillStyle = gradient;
```

要设置显示颜色，在渐变对象上使用 addColorStop 函数即可。除了可以变换成其他颜色外，还可以为颜色设置 Alpha 值，并且 Alpha 值也是可以变化的。为了达到这样的效果，需要使用颜色值的另一种表示方法，如内置 Alpha 组件的 CSSrgba 函数。绘制线性渐变时，会用到下面几个方法，如表 11-5 所示。

表 11-5　绘制线性渐变的方法

方　法	功　能
addColorStop	允许指定两个参数：颜色和偏移量。颜色参数是指开发人员希望在偏移位置描边或填充时所使用的颜色。偏移量是一个 0.0～1.0 的数值，代表沿着渐变线渐变的距离有多远
createLinearGradient(x0,y0,x1,y1)	沿着直线从(x0,y0)至(x1,y1)绘制渐变

【例 11.5】绘制线性渐变(案例文件：ch11\11.5.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>线性渐变</title>
</head>
<body>

<h1>绘制线性渐变</h1>
<canvas id="canvas" width="400" height="300"
  style="border:1px solid red"/>

<script type="text/javascript">
var c = document.getElementById("canvas");
var cxt = c.getContext("2d");
var gradient = cxt.createLinearGradient(0,0,0,canvas.height);
gradient.addColorStop(0,'#fff');
gradient.addColorStop(1,'#000');
cxt.fillStyle = gradient;
cxt.fillRect(0,0,400,400);
</script>
</body>
</html>
```

上述代码使用 2D 环境对象产生了一个线性渐变对象，渐变的起始点是(0,0)，渐变的结束

点是(0,canvas.height)，然后使用 addColorStop 函数设置渐变颜色，最后将渐变填充到上下文环境的样式中。

在 IE 11.0 中浏览，效果如图 11-5 所示。可以看到，在网页中创建了一个垂直方向上的渐变，从上到下颜色逐渐变深。



图 11-5　线性渐变

11.3.2　案例 6——绘制径向渐变

径向渐变即放射性渐变，是指颜色会介于两个指定圆之间的锥形区域平滑变化。径向渐变与线性渐变使用的颜色终止点是一样的。如果要实现径向渐变，需要使用 createRadialGradient 方法。

createRadialGradient(x0,y0,r0,x1,y1,r1)方法表示沿着两个圆之间的锥面绘制渐变。其中前 3 个参数代表开始的圆，圆心为(x0,y0)，半径为 r0。后 3 个参数代表结束的圆，圆心为(x1,y1)，半径为 r1。

【例 11.6】绘制径向渐变(案例文件：ch11\11.6.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>径向渐变</title>
</head>
<body>
<h1>绘制径向渐变</h1>
<canvas id="canvas" width="400" height="300" style="border:1px solid red"/>
<script type="text/javascript">
var c = document.getElementById("canvas");
var cxt = c.getContext("2d");
var gradient = cxt.createRadialGradient(
  canvas.width/2,canvas.height/2,0,canvas.width/2,canvas.height/2,150);
gradient.addColorStop(0,'#fff');
gradient.addColorStop(1,'#000');
cxt.fillStyle = gradient;
```

```
cxt.fillRect(0,0,400,400);
</script>
</body>
</html>
```

在上述代码中，首先创建渐变对象 gradient，此处使用 createRadialGradient 方法创建了一个径向渐变，然后使用 addColorStop 添加颜色，最后将渐变填充到上下文环境中。

在 IE 11.0 中浏览，效果如图 11-6 所示。可以看到，在网页中，从圆的中心亮点开始，向外逐步发散，形成了一个径向渐变。

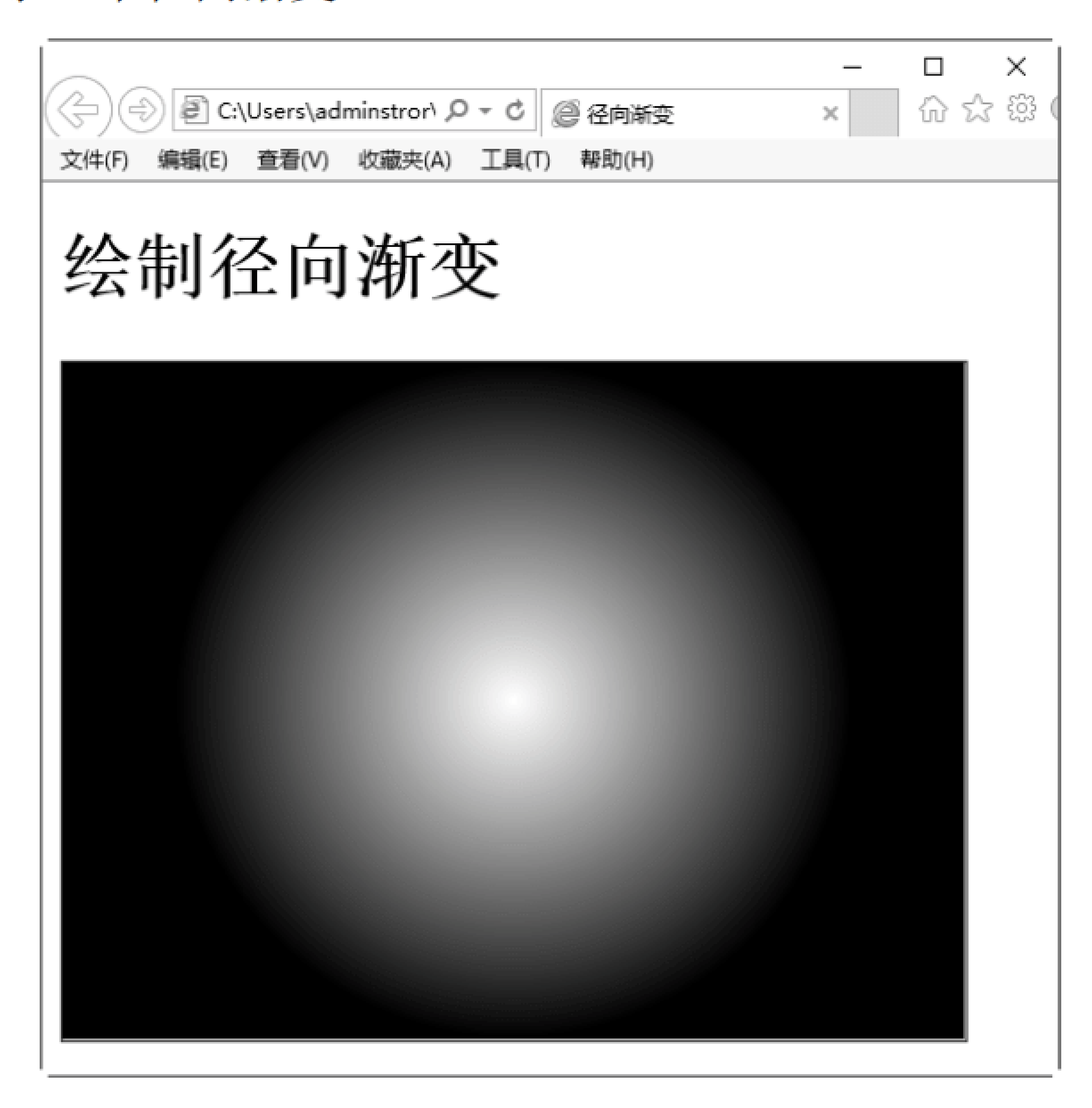


图 11-6　径向渐变

11.4　绘制变形图形

画布 canvas 不但可以使用 moveTo 这样的方法来移动画笔，来绘制图形和线条，还可以使用变换来调整画笔下的画布，变换的方法包括：平移、缩放、旋转、组合和带缩影等。

11.4.1　案例 7——绘制平移效果的图形

如果要对图形实现平移，需要使用 translate(x,y)方法，该方法表示在平面上平移，即原来原点为参考，然后以偏移后的位置作为坐标原点。也就是说，原来在(100,100)，然后 translate(1,1)，新的坐标原点在(101,101)而不是(1,1)。

【例 11.7】绘制平移效果的图形(案例文件：ch11\11.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制坐标变换</title>
```

```
<script>
function draw(id)
{
    var canvas = document.getElementById(id);
    if(canvas==null)
        return false;
    var context = canvas.getContext('2d');
    context.fillStyle = "#eeeeff";
    context.fillRect(0,0,400,300);
    context.translate(200,50);
    context.fillStyle = 'rgba(255,0,0,0.25)';
    for(var i=0; i<50; i++){
        context.translate(25,25);
        context.fillRect(0,0,100,50);
    }
}
</script>
</head>
<body onload="draw('canvas');">
<h1>变换原点坐标</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在 draw()函数中，使用 fillRect 方法绘制了一个矩形，然后使用 translate 方法平移到一个新位置，并从新位置开始，使用 for 循环，连续移动多次坐标原点，即多次绘制矩形。

在 IE 11.0 中浏览，效果如图 11-7 所示。可以看到，在网页中从坐标位置(200,50)开始绘制矩形，并每次以指定的平移距离绘制矩形。



图 11-7　变换原点坐标

11.4.2　案例 8——绘制缩放效果的图形

对变形图形来说，其中最常用的方式就是对图形进行缩放，即以原来的图形为参考，放

大或者缩小图形，从而增加效果。

如果要实现图形缩放，需要使用 scale(x,y)函数，该函数带有两个参数，分别代表在 x,y 两个方向上的值。每个参数在 canvas 显示图像的时候，向其传递在本方向轴上图像要放大(或者缩小)的量。如果 x 值为 2，就代表所绘制的图像中全部元素都会变成 2 倍宽。如果 y 值为 0.5，绘制出来的图像全部元素都会变成先前的一半高。

【例 11.8】绘制缩放效果的图形(案例文件：ch11\11.8.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图形缩放</title>
<script>
function draw(id)
{
   var canvas = document.getElementById(id);
   if(canvas==null)
      return false;
   var context = canvas.getContext('2d');
   context.fillStyle = "#eeeeff";
   context.fillRect(0,0,400,300);
   context.translate(200,50);
   context.fillStyle = 'rgba(255,0,0,0.25)';
   for(var i=0; i<50; i++){
      context.scale(3,0.5);
      context.fillRect(0,0,100,50);
   }
}
</script>
</head>
<body onload="draw('canvas');">
<h1>图形缩放</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在上述代码中，缩放操作是放在 for 循环中完成的，在此循环中，以原来图形为参考物，使其在 x 轴方向增加为 3 倍宽，在 y 轴方向上变为原来的一半。

在 IE 11.0 中浏览，效果如图 11-8 所示。可以看到，在一个指定方向上绘制了多个矩形。

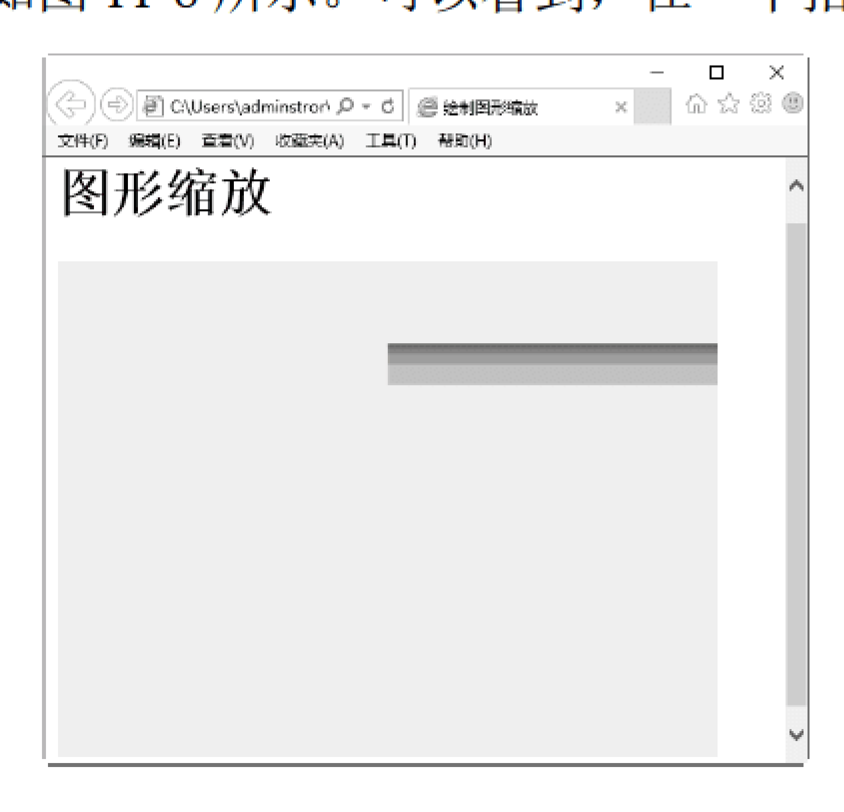


图 11-8　图形缩放

11.4.3 案例 9——绘制旋转效果的图形

变换操作并不限于平移和缩放，还可以使用函数 context.rotate(angle)来旋转图像，甚至可以直接修改底层变换矩阵以完成一些高级操作，如剪裁图像的绘制路径。

例如，context.rotate(1.57)表示旋转角度参数以弧度为单位。rotate()方法默认地从左上端的(0,0)开始旋转，通过指定一个角度，改变了画布坐标和 Web 浏览器中的<canvas>元素的像素之间的映射，使得任意后续绘图在画布中都显示为旋转的。

【例 11.9】绘制旋转效果的图形(案例文件：ch11\11.9.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制旋转图像</title>
<script>
function draw(id)
{
   var canvas = document.getElementById(id);
   if(canvas==null)
      return false;
   var context = canvas.getContext('2d');
   context.fillStyle = "#eeeeff";
   context.fillRect(0,0,400,300);
   context.translate(200,50);
   context.fillStyle = 'rgba(255,0,0,0.25)';
   for(var i=0; i<50; i++){
      context.rotate(Math.PI/10);
      context.fillRect(0,0,100,50);
   }
}
</script>
</head>
<body onload="draw('canvas');">
<h1>旋转图形</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在上述代码中，使用 rotate 方法，在 for 循环中对多个图形进行了旋转，其旋转角度相同。在 IE 11.0 中浏览，效果如图 11-9 所示。可以看到，多个矩形以中心弧度为原点进行了旋转。

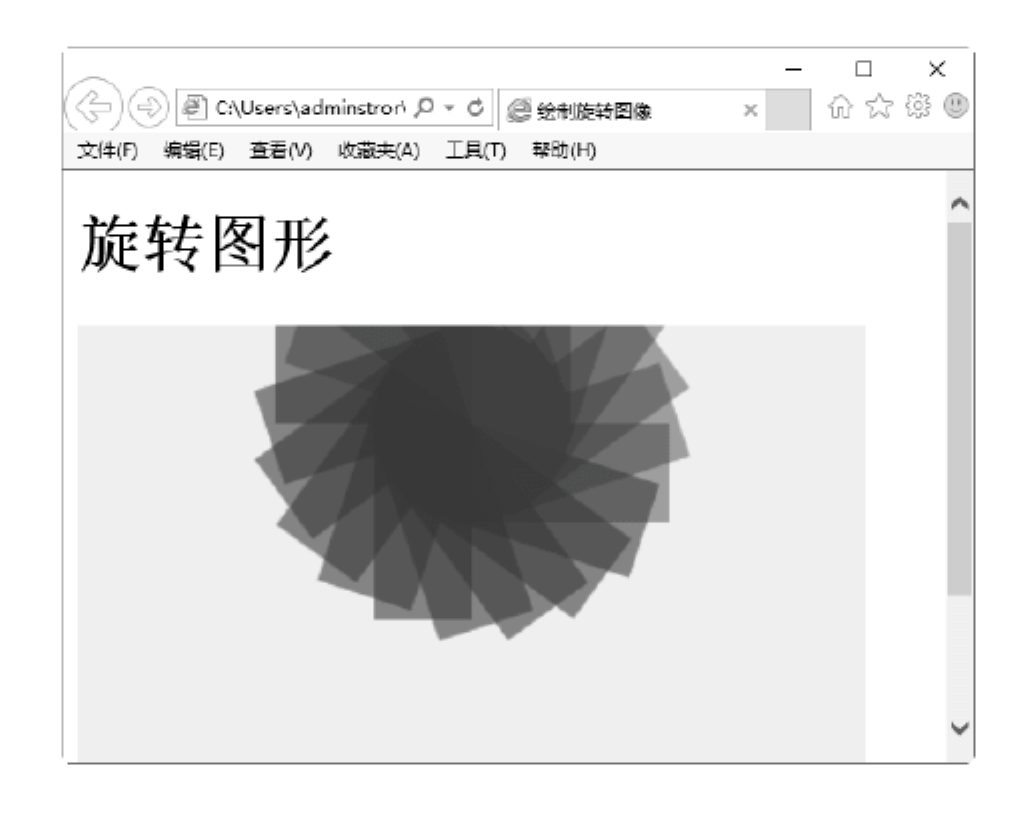


图 11-9 旋转图形


注意

这个操作并没有旋转<canvas>元素本身。而且，旋转的角度是用弧度指定的。

11.4.4 案例 10——绘制组合效果的图形

在前面介绍的知识中，可以将一个图形画在另一个之上。在大多数情况下，这样是不够的。例如，这样会受制于图形的绘制顺序。不过，我们可以利用 globalCompositeOperation 属性来改变这些做法，不仅可以在已有图形后面再画新图形，还可以用来遮盖、清除(比 clearRect 方法强劲得多)某些区域。

其语法格式如下：

```
globalCompositeOperation = type
```

这表示设置不同形状的组合类型，其中 type 表示方的图形是已经存在的 canvas 内容，圆的图形是新的形状，其默认值为 source-over，表示在 canvas 内容上面画新的形状。

type 具有 12 个属性值，具体说明如表 11-6 所示。

表 11-6 type 的属性值

属 性 值	说 明
source-over(default)	这是默认设置，新图形会覆盖在原有内容之上
destination-over	会在原有内容之下绘制新图形
source-in	新图形会仅仅出现在与原有内容重叠的部分，其他区域都变成透明的
destination-in	原有内容中与新图形重叠的部分会被保留，其他区域都变成透明的
source-out	结果是只有新图形中与原有内容不重叠的部分会被绘制出来
destination-out	原有内容中与新图形不重叠的部分会被保留
source-atop	新图形中与原有内容重叠的部分会被绘制，并覆盖于原有内容之上
destination-atop	原有内容中与新内容重叠的部分会被保留，并会在原有内容之下绘制新图形
lighter	两图形中重叠的部分做加色处理
darker	两图形中重叠的部分做减色处理
xor	重叠的部分会变成透明
copy	只有新图形会被保留，其他都被清除掉

【例 11.10】绘制组合效果的图形(案例文件：ch11\11.10.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图形组合</title>
<script>
function draw(id)
{
    var canvas = document.getElementById(id);
    if(canvas==null)
        return false;
```

```
    var context = canvas.getContext('2d');
    var oprtns = new Array(
        "source-atop",
        "source-in",
        "source-out",
        "source-over",
        "destination-atop",
        "destination-in",
        "destination-out",
        "destination-over",
        "lighter",
        "copy",
        "xor"
    );
    var i = 10;
    context.fillStyle = "blue";
    context.fillRect(10,10,60,60);
    context.globalCompositeOperation = oprtns[i];
    context.beginPath();
    context.fillStyle = "red";
    context.arc(60,60,30,0,Math.PI*2,false);
    context.fill();
}
</script>
</head>
<body onload="draw('canvas');">
<h1>图形组合</h1>
<canvas id="canvas" width="400" height="300" />
</body>
</html>
```

在上述代码中，首先创建了一个 oprtns 数组，用于存储 type 的 12 个值，然后绘制了一个矩形，并使用 content 上下文对象设置了图形的组合方式，即采用新图形显示、其他被清除的方式，最后使用 arc 绘制了一个圆。

在 IE 11.0 中浏览，效果如图 11-10 所示。可以看到，页面上绘制了一个矩形和圆，但矩形和圆接触的地方以空白显示。



图 11-10 图形组合

11.4.5　案例 11——绘制带阴影的图形

在画布 canvas 上绘制带有阴影效果的图形非常简单，只需要设置几个属性即可。这些属性分别为 shadowOffsetX、shadowOffsetY、shadowBlur 和 shadowColor。

属性 shadowColor 表示阴影的颜色，其值与 CSS 颜色值一致。shadowBlur 表示设置阴影模糊程度，此值越大，阴影越模糊。shadowOffsetX 和 shadowOffsetY 属性表示阴影的 x 和 y 偏移量，单位是像素。

【例 11.11】绘制带阴影的图形(案例文件：ch11\11.11.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制阴影效果图形</title>
</head>
<body>
<canvas id="my canvas" width="200" height="200"
  style="border:1px solid #ff0000">
</canvas>
<script type="text/javascript">
var elem = document.getElementById("my canvas");
if (elem && elem.getContext)  {
    var context = elem.getContext("2d");
    //shadowOffsetX 和 shadowOffsetY：阴影的 x 和 y 偏移量，单位是像素
    context.shadowOffsetX = 15;
    context.shadowOffsetY = 15;
    //shadowBlur：设置阴影模糊程度。此值越大，阴影越模糊
    //其效果与 Photoshop 的高斯模糊滤镜相同
    context.shadowBlur = 10;
    //shadowColor：阴影颜色。其值与 CSS 颜色值一致
    //context.shadowColor = 'rgba(255, 0, 0, 0.5)';  或下面的十六进制表示法
    context.shadowColor = '#f00';
    context.fillStyle = '#00f';
    context.fillRect(20, 20, 150, 100);
}
</script>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 11-11 所示。可以看到，页面上显示了一个蓝色矩形，其阴影为红色矩形。

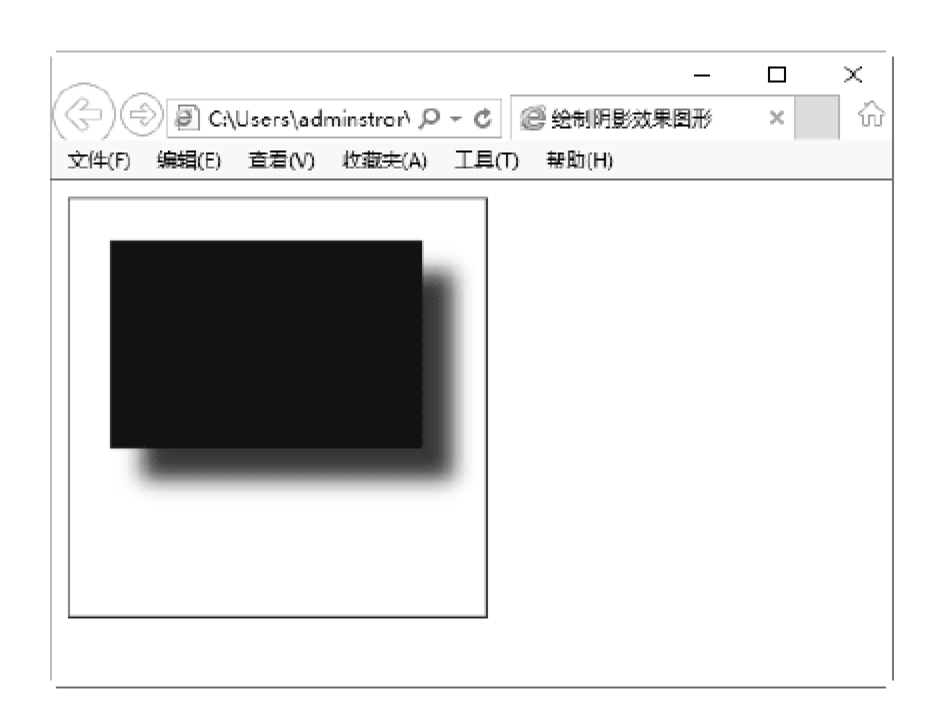


图 11-11　带有阴影的图形



11.5 使 用 图 像

画布 canvas 有一项功能就是可以引入图像，以便用于图片合成或者制作背景等。而目前仅可以在图像中加入文字。只要是 Geck 支持的图像(如 PNG、GIF、JPEG 等)都可以引入到 canvas 中，而且其他的 canvas 元素也可以作为图像的来源。

11.5.1 案例 12——绘制图像

要在画布 canvas 上绘制图像，需要先有一张图片。这张图片可以是已经存在的<img>元素，或者通过 JS 创建。

无论采用哪种方式，都需要在绘制 canvas 之前完全加载这张图片。浏览器通常会在页面脚本执行的同时异步加载图片。如果试图在图片未完全加载之前就将其呈现到 canvas 上，那么 canvas 将不会显示任何图片。

捕获和绘制图像完全是通过 drawImage()方法完成的，它可以接受不同的 HTML 参数，具体含义如表 11-7 所示。

表 11-7 绘制图像的方法

方 法	说 明
drawIamge(image,dx,dy)	接受一张图片，并将其画到 canvas 中。给出的坐标(dx,dy)代表图片的左上角。例如，坐标(0,0)将把图片画到 canvas 的左上角
drawIamge(image,dx,dy,dw,dh)	接受一张图片，将其缩放为宽度 dw 和高度 dh，然后把它画到 canvas 上的(dx,dy)位置
drawIamge(image,sx,sy,sw,sh,dx,dy,dw,dh)	接受一张图片，通过参数(sx,sy,sw,sh)指定图片裁剪的范围，缩放到(dw,dh)的大小，最后把它画到 canvas 上的(dx,dy)位置

【例 11.12】绘制图像(案例文件：ch11\11.12.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图像</title>
</head>
<body>
<canvas id="canvas" width="300" height="200" style="border:1px solid blue">
   Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
window.onload=function(){
   var ctx = document.getElementById("canvas").getContext("2d");
   var img = new Image();
   img.src = "01.jpg";
   img.onload=function(){
```

```
        ctx.drawImage(img,0,0);
    }
}
</script>
</body>
</html>
```

在上述代码中，使用窗口的 onload 加载事件，即在页面被加载时执行函数。在函数中，创建上下文对象 ctx，并创建 Image 对象 img；然后使用 img 对象的 src 属性设置图片来源，最后使用 drawImage 画出当前的图像。

在 IE 11.0 中浏览，效果如图 11-12 所示。可以看到，页面上绘制了一个图像，并且在画布中显示。



图 11-12 绘制图像

11.5.2 案例 13——平铺图像

使用画布 canvas 绘制图像有很多种用处，其中一个用处就是将绘制的图像作为背景图片使用。在做背景图片时，如果显示图片的区域大小不能直接设定，通常将图片以平铺的方式显示。

HTML 5 Canvas API 支持图片平铺，此时需要调用 createPattern 函数，即调用 createPattern 函数来替代先前的 drawImage 函数。函数 createPattern 的语法格式如下：

```
createPattern(image,type)
```

其中 image 表示要绘制的图像。type 表示平铺的类型，其具体含义如表 11-8 所示。

表 11-8 type 表示平铺的类型

参 数 值	说 明
no-repeat	不平铺
repeat-x	横方向平铺
repeat-y	纵方向平铺
repeat	全方向平铺

【例 11.13】平铺图像(案例文件：ch11\11.13.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图像平铺</title>
</head>
<body onload="draw('canvas');">
<h1>图形平铺</h1>
<canvas id="canvas" width="800" height="600"></canvas>
<script>
function draw(id){
   var canvas = document.getElementById(id);
   if(canvas==null){
      return false;
   }
   var context = canvas.getContext('2d');
   context.fillStyle = "#eeeeff";
   context.fillRect(0,0,800,600);
   image = new Image();
   image.src = "02.jpg";
   image.onload = function(){
      var ptrn = context.createPattern(image,'repeat');
      context.fillStyle = ptrn;
      context.fillRect(0,0,800,600);
   }
}
</script>
</body>
</html>
```

在上述代码中，用 fillRect 创建了一个宽度为 800、高度为 600，左上角坐标位置为(0,0)的矩形，然后创建了一个 Image 对象，src 表示链接一个图像源，然后使用 createPattern 绘制一个图像，其方式是完全平铺，并将这个图像作为一个模式填充到矩形中。最后绘制这个矩形，此矩形的大小完全覆盖原来的图形。

在 IE 11.0 中浏览，效果如图 11-13 所示。可以看到，页面上绘制了一个图像，其图像以平铺的方式充满整个矩形。



图 11-13 图像平铺

11.5.3 案例 14——裁剪图像

要完成对图像的裁剪，需要用到 clip 方法。clip 方法表示给 canvas 设置一个剪辑区域。在调用 clip 方法之后，所有代码只对这个设定的剪辑区域有效，不会影响其他地方。这个方法在要进行局部更新时很有用。在默认情况下，剪辑区域是一个左上角在(0,0)，宽和高分别等于 canvas 元素的宽和高的矩形。

【例 11.14】裁剪图像(案例文件：ch11\11.14.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图像裁剪</title>
<script type="text/javascript" src="script.js"></script>
</head>
<body onload="draw('canvas');">
<h1>图像裁剪实例</h1>
<canvas id="canvas" width="400" height="300"></canvas>
<script>
function draw(id){
    var canvas = document.getElementById(id);
    if(canvas==null){
        return false;
    }
    var context = canvas.getContext('2d');
    var gr = context.createLinearGradient(0,400,300,0);
    gr.addColorStop(0,'rgb(255,255,0)');
    gr.addColorStop(1,'rgb(0,255,255)');
    context.fillStyle = gr;
    context.fillRect(0,0,400,300);
    image = new Image();
    image.onload=function(){
        drawImg(context,image);
    };
    image.src = "02.jpg";
}
function drawImg(context,image){
    create8StarClip(context);
    context.drawImage(image,-50,-150,300,300);
}
function create8StarClip(context){
    var n = 0;
    var dx = 100;
    var dy = 0;
    var s = 150;
    context.beginPath();
    context.translate(100,150);
    var x = Math.sin(0);
    var y = Math.cos(0);
    var dig = Math.PI/5*4;
    for(var i=0; i<8; i++){
        var x = Math.sin(i*dig);
        var y = Math.cos(i*dig);
        context.lineTo(dx+x*s,dy+y*s);
    }
    context.clip();
}
</script>
</body>
</html>
```

在上述代码中，创建了 3 个 JavaScript 函数，其中 create8StarClip 函数完成了多边的图形创建，以此图形作为裁剪的依据。drawImg 函数表示绘制一个图形，其图形带有裁剪区域。draw 函数完成对画布对象的获取，并定义一个线性渐变，然后创建了一个 Image 对象。

在 IE 11.0 中浏览，效果如图 11-14 所示。可以看到，页面上绘制了一个五边形，图像作为五边形的背景显示，从而实现了对图像的裁剪。



图 11-14 图像裁剪

11.5.4 案例 15——图像的像素化处理

在画布中，可以使用 ImageData 对象来保存图像的像素值，它有 width、height 和 data 这 3 个属性，其中 data 属性就是一个连续数组。图像的所有像素值其实是保存在 data 里面的。

data 属性保存像素值的方法如下：

```
imageData.data[index*4+0]
imageData.data[index*4+1]
imageData.data[index*4+2]
imageData.data[index*4+3]
```

上面取出了 data 数组中连续相邻的 4 个值，这 4 个值分别代表了图像中第 index+1 个像素的红色、绿色、蓝色和透明度值的大小。需要注意的是，index 从 0 开始，图像中总共有 width*height 个像素，数组中总共保存了 width*height*4 个数值。

画布对象有 3 个方法，用来创建、读取和设置 ImageData 对象，如表 11-9 所示。

表 11-9 创建画布对象的方法

方 法	说 明
createImageData(width, height)	在内存中创建一个指定大小的 ImageData 对象(即像素数组)，对象中的像素点都是黑色透明的，即 rgba(0,0,0,0)
getImageData(x, y, width, height)	返回一个 ImageData 对象，这个 ImageData 对象中包含了指定区域的像素数组
putImageData(data, x, y)	将 ImageData 对象绘制到屏幕的指定区域上

【例 11.15】图像的像素化处理(案例文件：ch11\11.15.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>图像像素处理</title>
<script type="text/javascript" src="script.js"></script>
</head>
<body onload="draw('canvas');">
<h1>像素处理示例</h1>
<canvas id="canvas" width="400" height="300"></canvas>
<script>
function draw(id){
    var canvas = document.getElementById(id);
    if(canvas==null){
        return false;
    }
    var context = canvas.getContext('2d');
    image = new Image();
    image.src = "01.jpg";
    image.onload=function(){
        context.drawImage(image,0,0);
        var imagedata = context.getImageData(0,0,image.width,image.height);
        for(var i=0,n=imagedata.data.length; i<n; i+=4){
            imagedata.data[i+0] = 255-imagedata.data[i+0];
            imagedata.data[i+1] = 255-imagedata.data[i+2];
            imagedata.data[i+2] = 255-imagedata.data[i+1];
        }
        context.putImageData(imagedata,0,0);
    };
}
</script>
</body>
</html>
```

在上述代码中，使用 getImageData()方法获取一个 ImageData 对象，并包含相关的像素数组。在 for 循环中，对像素值重新赋值，最后使用 putImageData 将处理过的图像在画布上绘制出来。

在 IE 11.0 中浏览，效果如图 11-15 所示。可以看到，页面上显示了一个图像，其图像明显经过像素处理，显示得没有原来清晰。

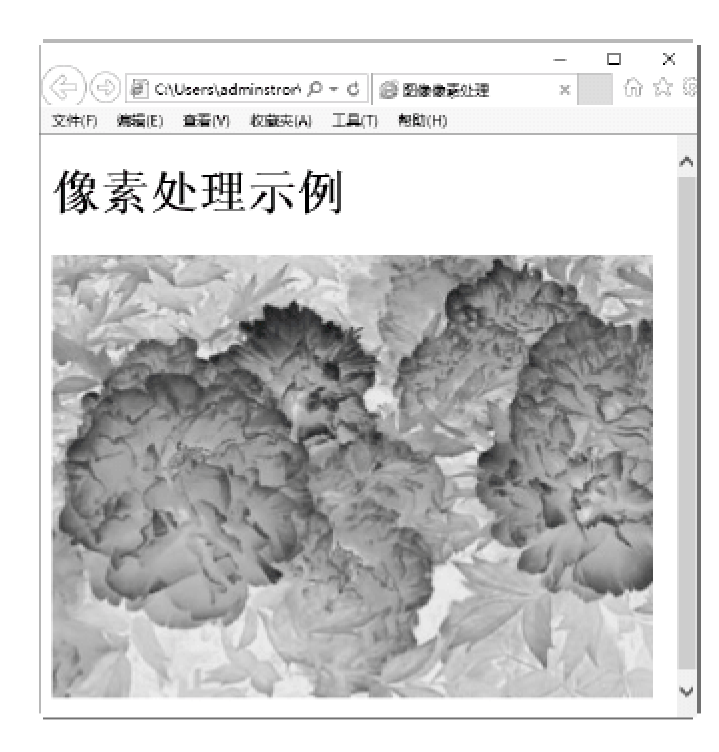


图 11-15　像素处理

11.6 案例 16——绘制文字

在画布中绘制字符串(文字)的方式，与操作其他路径对象的方式相同，可以描绘文本轮廓和填充文本内部，同时，所有能够应用于其他图形的变换和样式都能用于文本。

文本绘制功能涉及的方法如表 11-10 所示。

表 11-10 绘制文本的方法

方 法	说 明
fillText(text,x,y,maxwidth)	绘制带 fillStyle 填充的文字，拥有文本参数以及用于指定文本位置的坐标的参数。maxwidth 是可选参数，用于限制字体大小，它会将文本字体强制收缩到指定尺寸
trokeText(text,x,y,maxwidth)	绘制只有 strokeStyle 边框的文字，其参数含义与上一个方法相同
measureText	该函数会返回一个度量对象，它包含了在当前 context 环境下指定文本的实际显示宽度

为了保证文本在各浏览器中都能正常显示，在绘制上下文里有以下字体属性。

(1) font。可以是 CSS 字体规则中的任何值。包括字体样式、字体变种、字体大小与粗细、行高和字体名称。

(2) textAlign。控制文本的对齐方式。它类似于(但不完全等同于)CSS 中的 text-align。可能的取值为 start、end、left、right 和 center。

(3) textBaseline。控制文本相对于起点的位置。可以取值为 top、hanging、middle、alphabetic、ideographic 和 bottom。对于简单的英文字母，可以放心地使用 top、middle 或 bottom 作为文本基线。

【例 11.16】绘制文字(案例文件：ch11\11.16.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas</title>
</head>
<body>
<canvas id="my canvas" width="200" height="200"
  style="border:1px solid #ff0000">
</canvas>
<script type="text/javascript">
var elem = document.getElementById("my_canvas");
if (elem && elem.getContext) {
    var context = elem.getContext("2d");
    context.fillStyle = '#00f';
    //font: 文字字体，同 CSSfont-family 属性
    context.font = 'italic  30px 微软雅黑';    //斜体 30 像素，微软雅黑字体
    //textAlign: 文字水平对齐方式。
    //可取属性值: start, end, left, right, center。默认值:start
    context.textAlign = 'left';
```

```
    //文字竖直对齐方式。
    //可取属性值：top, hanging, middle, alphabetic, ideographic, bottom
    //默认值：alphabetic
    context.textBaseline = 'top';
    //要输出的文字内容，文字位置坐标，第 4 个参数为可选选项——最大宽度。
    //如果需要的话，浏览器会缩减文字，以让它适应指定宽度
    context.fillText('生日快乐!', 0, 0,50);    //有填充
    context.font = 'bold 30px sans-serif';
    context.strokeText('生日快乐!', 0, 50,100);  //只有文字边框
}
</script>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 11-16 所示。可以看到，页面上显示了一个画布边框，画布中显示了两个不同的字符串。第一个字符串以斜体显示，其颜色为蓝色。第二个字符串字体颜色为浅黑色，加粗显示。



图 11-16　绘制文字

11.7　图形的保存与恢复

在画布对象中绘制图形或图像时，可以将这些图形或者图形的状态进行保存，即永久保存图形或图像。

11.7.1　案例 17——保存与恢复状态

在画布对象中，由两个方法管理绘制状态的当前栈，save 方法把当前状态压入栈中，而 restore 方法从栈顶弹出状态。绘制状态不会覆盖对画布所做的每件事情。其中 save 方法用来保存 canvas 的状态。save 之后，可以调用 canvas 的平移、缩放、旋转、错切、裁剪等操作。restore 方法用来恢复 canvas 先前保存的状态，防止 save 后对 canvas 执行的操作对后续的绘制有影响。save 和 restore 要配对使用(restore 可以比 save 少，但不能多)，如果 restore 调用次数

比 save 多，会引发 Error。

【例 11.17】保存与恢复状态(案例文件：ch11\11.17.html)。

```
<!DOCTYPE html>
<html>
<head><title>保存与恢复</title></head>
<body>
<canvas id="myCanvas" width="500" height="400"
  style="border:1px solid blue">
   Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "rgb(0,0,255)";
ctx.save();
ctx.fillRect(50,50,100,100);
ctx.fillStyle = "rgb(255,0,0)";
ctx.save();
ctx.fillRect(200,50,100,100);
ctx.restore();
ctx.fillRect(350,50,100,100);
ctx.restore();
ctx.fillRect(50, 200, 100, 100);
</script>
</body>
</html>
```

在上述代码中，绘制了 4 个矩形，在第 1 个矩形绘制之前，定义当前矩形的显示颜色，并将此样式加入到栈中，然后创建了一个矩形。在第 2 个矩形绘制之前，重新定义了矩形显示颜色，并使用 save 将此样式压入到栈中，然后创建了一个矩形。在第 3 个矩形绘制之前，使用 restore 恢复当前显示颜色，即调用栈中的最上层颜色，绘制矩形。在第 4 个矩形绘制之前，继续使用 restore 方法，调用最后一个栈中的元素来定义矩形颜色。

在 IE 11.0 中浏览，效果如图 11-17 所示。可以看到，页面上绘制了 4 个矩形，第 1 个和第 4 个矩形显示为蓝色，第 2 个和第 3 个矩形显示为红色。

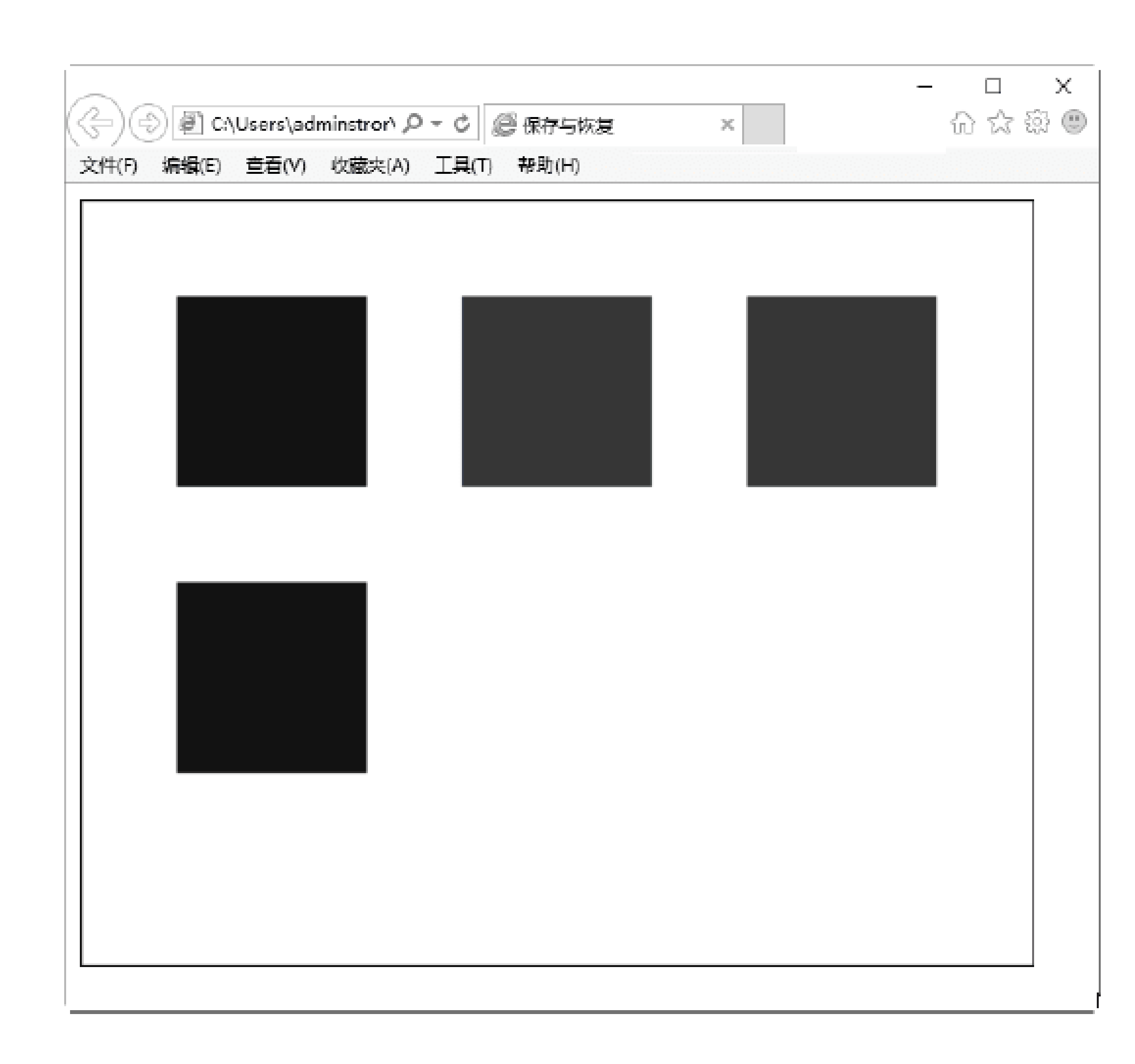


图 11-17　保存与恢复

11.7.2　案例 18——保存文件

当绘制出漂亮的图形时，有时需要保存这些劳动成果。这时可以将当前的画布元素(而不

是 2D 环境)的当前状态导出到数据 URL。导出很简单，可以利用 toDataURL 方法来完成，它可以调用不同的图片格式。目前 Firefox 和 Opera 浏览器只支持 PNG 格式，Safari 支持 GIF、PNG 和 JPG 格式。大多数浏览器支持读取 base64 编码内容，如一幅图像。URL 的语法格式如下：

```
data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAfQAAAH0CAYAAADL1t
```

它以一个 data 开始，然后是 mine 类型，之后是编码和 base64，最后是原始数据。这些原始数据就是画布元素所要导出的内容，并且浏览器能够将数据编码为真正的资源。

【例 11.18】保存文件(案例文件：ch11\11.18.html)。

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="500" height="500"
  style="border:1px solid blue">
    Your browser does not support the canvas element.
</canvas>
<script type="text/javascript">
var c = document.getElementById("myCanvas");
var cxt = c.getContext("2d");
cxt.fillStyle = 'rgb(0,0,255)';
cxt.fillRect(0,0,cxt.canvas.width,cxt.canvas.height);
cxt.fillStyle = "rgb(0,255,0)";
cxt.fillRect(10,20,50,50);
window.location = cxt.canvas.toDataURL('image/png');
</script>
</body>
</html>
```

在上述代码中，使用 canvas.toDataURL 语句将当前绘制图像保存到 URL 数据中。在 Firefox 53.0 中浏览，效果如图 11-18 所示。可以看到，在浏览器的地址栏中显示的是 URL 数据。

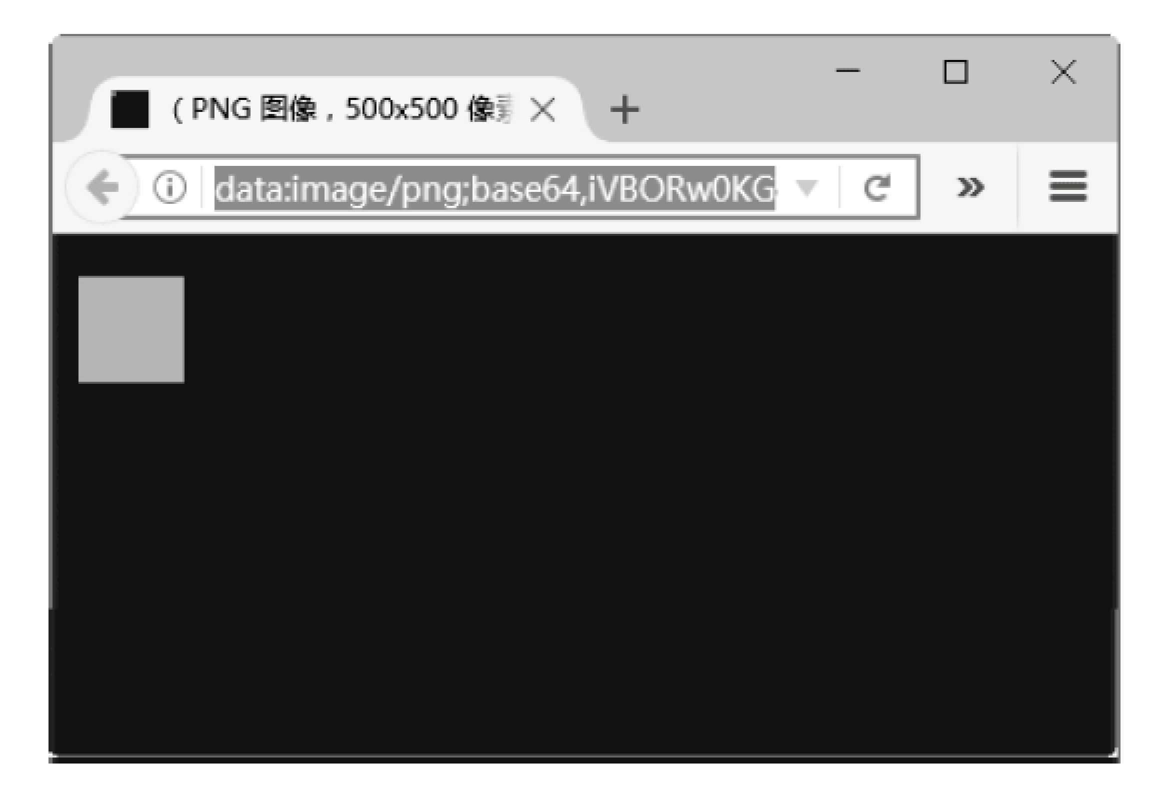


图 11-18　保存图形

11.8 综合案例——绘制火柴棒人物

漫画中最常见的一种图形，就是火柴棒人，通过简单的几个笔画，就可以绘制出一个传神的动漫人物。使用 canvas 和 JavaScript 同样可以绘制一个火柴棒人物。具体操作步骤如下。

step 01 分析需求。

一个火柴棒人，由两个部分组成：脸部和身躯。脸部是一个圆形，其中包括眼睛和嘴；身躯由几条直线组成，包括胳膊和腿等。实际上此案例就是绘制圆形、弧度和直线的组合。示例完成后，效果如图 11-19 所示。

step 02 实现 HTML 页面，定义画布 canvas：

```
<!DOCTYPE html>
<html>
<title>绘制火柴棒人</title>
<body>
<canvas id="myCanvas" width="500" height="300"
  style="border:1px solid blue">
   Your browser does not support the canvas element.
</canvas>
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 11-20 所示，页面中显示了一个画布边框。



图 11-19　火柴棒人

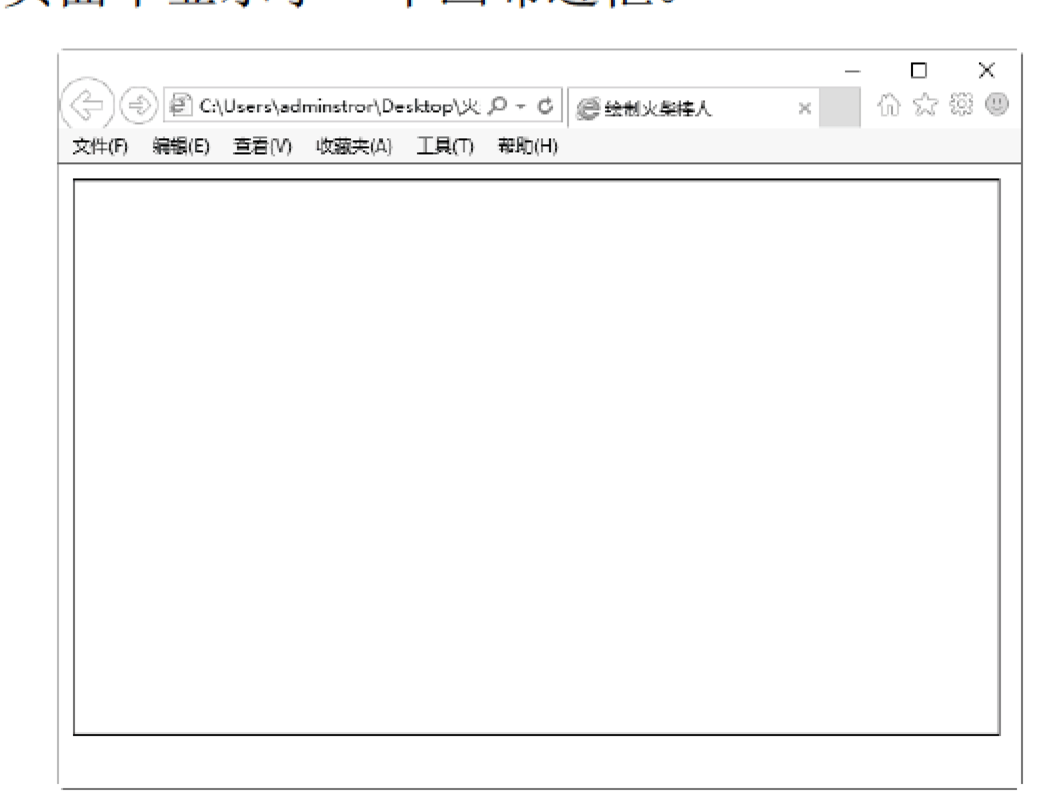


图 11-20　定义画布边框

step 03 实现头部轮廓绘制：

```
<script type="text/javascript">
var c = document.getElementById("myCanvas");
var cxt = c.getContext("2d");
cxt.beginPath();
cxt.arc(100,50,30,0,Math.PI*2,true);
cxt.fill();
</script>
```

这会产生一个实心的、填充的头部，即圆形。在 arc 函数中，x 和 y 的坐标为(100,50)，

半径为 30 像素，另两个参数为弧度的开始和结束，第 6 个参数表示绘制弧形的方向，即顺时针和逆时针方向。

在 IE 11.0 中浏览，效果如图 11-21 所示，页面中显示了实心圆，其颜色为黑色。

step 04 用 JS 绘制笑脸：

```
cxt.beginPath();
cxt.strokeStyle = '#c00';
cxt.lineWidth = 3;
cxt.arc(100,50,20,0,Math.PI,false);
cxt.stroke();
```

此处使用 beginPath 方法，表示重新绘制，并设定线条宽度，然后绘制了一个弧形，这个弧形是从嘴部开始的弧形。

在 IE 11.0 中浏览，效果如图 11-22 所示，页面中显示了一个漂亮的半圆式的笑脸。

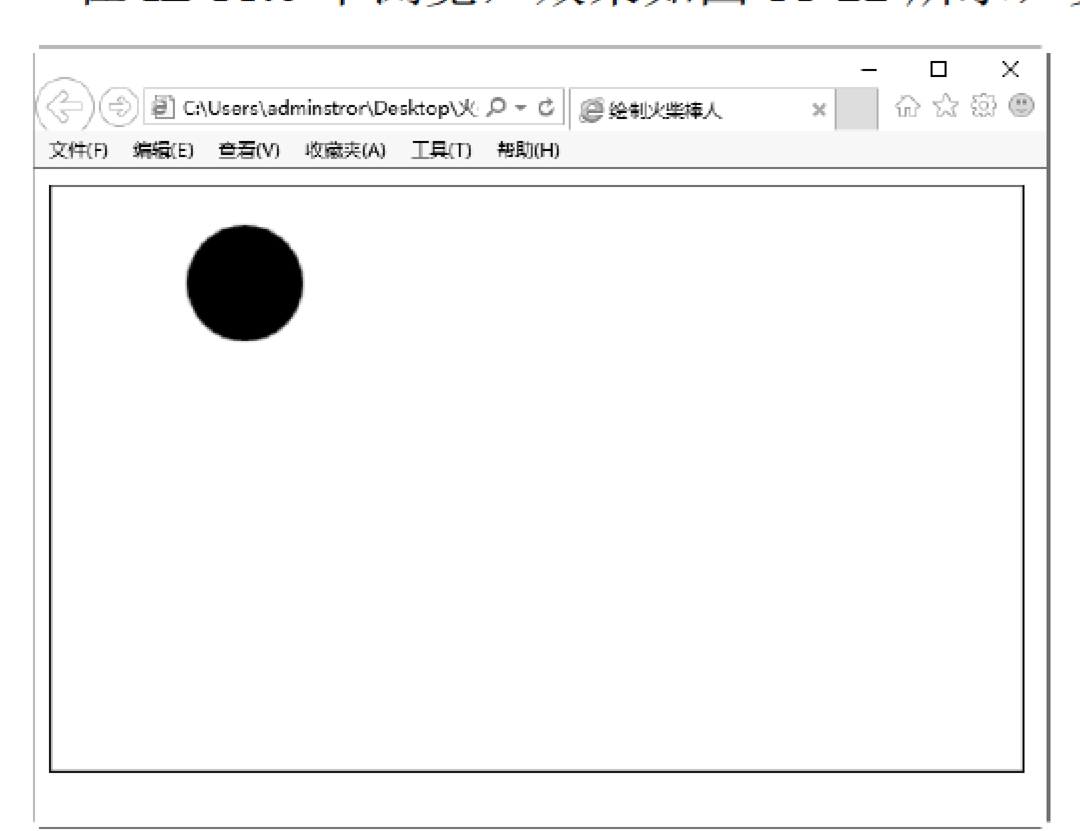


图 11-21　绘制头部轮廓

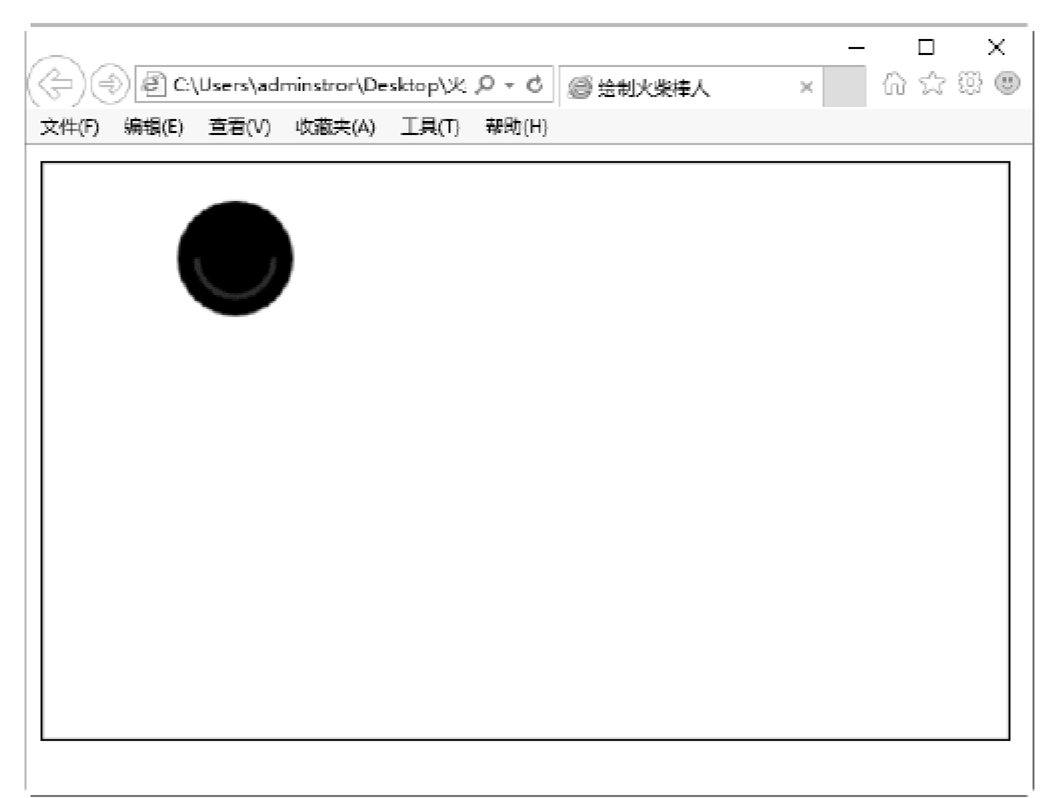


图 11-22　绘制笑脸

step 05 绘制眼睛：

```
cxt.beginPath();
cxt.fillStyle = "#c00";
cxt.arc(90,45,3,0,Math.PI*2,true);
cxt.fill();
cxt.moveTo(113,45);
cxt.arc(110,45,3,0,Math.PI*2,true);
cxt.fill();
cxt.stroke();
```

首先填充弧线，创建了一个实体样式的眼睛，arc 绘制左眼，然后使用 moveTo 绘制右眼。在 IE 11.0 中浏览，效果如图 11-23 所示，页面中显示了一双眼睛。

step 06 绘制身躯：

```
cxt.moveTo(100,80);
cxt.lineTo(100,150);
cxt.moveTo(100,100),
cxt.lineTo(60,120);
cxt.moveTo(100,100);
cxt.lineTo(140,120);
```

```
cxt.moveTo(100,150);
cxt.lineTo(80,190);
cxt.moveTo(100,150);
cxt.lineTo(140,190);
cxt.stroke();
```

上述代码以 moveTo 作为开始坐标，以 lineTo 为终点，绘制不同的直线，这些直线的坐标位置需要在不同地方汇集，两只手在坐标位置(100,100)处交叉，两只脚在坐标位置(100,150)处交叉。

在 IE 11.0 中浏览，效果如图 11-24 所示，页面中显示了一个火柴棒人，与上一个图形相比，多了一个身躯。



图 11-23　绘制眼睛



图 11-24　绘制身躯

11.9　跟我学上机——绘制商标

绘制商标是 canvas 画布的用途之一，可以绘制 adidas 和 nike 商标。nike 的图标比 adidas 的复杂得多，adidas 都是直线组成，而 nike 的多了曲线。实现本例的具体操作步骤如下。

step 01 分析需求。

要绘制两条曲线，需要找到曲线的参考点(参考点决定了曲线的曲率)，这需要慢慢地移动，然后再看效果，反反复复。quadraticCurveTo(30,79,99,78)函数有两组坐标：第一组坐标为控制点，决定曲线的曲率；第二组坐标为终点。

step 02 构建 HTML，实现 canvas 画布：

```
<!DOCTYPE html>
<html>
<head>
<title>绘制商标</title>
</head>
<body>
<canvas id="nike" width="375px" height="132px"
  style="border:1px solid #000;">
</canvas>
```

```
</body>
</html>
```

在 IE 11.0 中浏览，效果如图 11-25 所示，页面中只显示了一个画布边框，其内容还没有绘制。

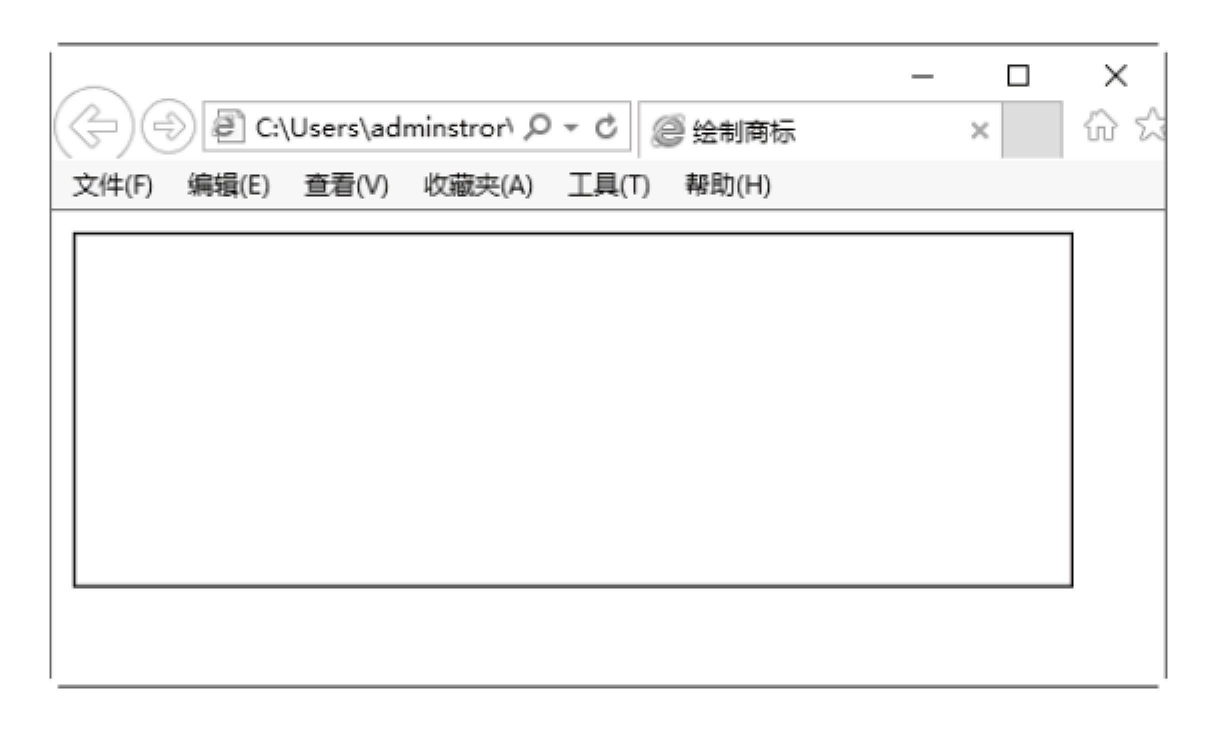


图 11-25　定义画布边框

step 03 用 JS 实现基本图形：

```
<script>
function drawNike(){
    //取得 canvas 元素及其绘图上下文
    var canvas = document.getElementById('Nike');
    var context = canvas.getContext('2d');
    //保存当前的绘图状态
    context.save();
    //开始绘制打钩的轮廓
    context.beginPath();
    context.moveTo(53,0);
    //绘制上半部分曲线，第一组坐标为控制点，决定曲线的曲率，第二组坐标为终点
    context.quadraticCurveTo(30,79,99,78);
    context.lineTo(371,2);
    context.lineTo(74,134);
    context.quadraticCurveTo(-55,124,53,0);
    //用红色填充
    context.fillStyle = "#da251c";
    context.fill();
    //用 3 像素深红线条描边
    context.lineWidth = 3;
    //连接处平滑
    context.lineJoin = 'round';
    context.strokeStyle = "#d40000";
    context.stroke();
    //恢复原有的绘图状态
    context.restore();
}
window.addEventListener("load",drawNike,true);
</script>
```

在 IE 11.0 中浏览，效果如图 11-26 所示，显示了一个商标图案，颜色为红色。



图 11-26 绘制商标

11.10 高 手 解 惑

疑问 1：canvas 的宽度和高度是否可以在 CSS 属性中定义呢？

答：添加 canvas 标签的时候，会在 canvas 的属性里填写要初始化的 canvas 的高度和宽度：

```
<canvas width="500" height="400">Not Supported!</canvas>
```

如果把高度和宽度写在了 CSS 里面，结果会发现，在绘图时坐标获取出现差异，canvas.width 和 canvas.height 分别是 300 和 150，与预期的不一样。这是因为 canvas 要求这两个属性必须随 canvas 标记一起出现。

疑问 2：画布中 Stroke 和 Fill 二者的区别是什么？

答：HTML 5 中，将图形分为两大类：第一类称作 Stroke，就是轮廓、勾勒或者线条，总之，图形是由线条组成的；第二类称作 Fill，就是填充区域。上下文对象中有两个绘制矩形的方法，可以让我们很好地理解这两大类图形的区别：一是 strokeRect；二是 fillRect。

第 III 篇

高 级 技 能

- 第 12 章　HTML 5 中的文件与拖放
- 第 13 章　定位地理位置技术
- 第 14 章　Web 存储和通信技术
- 第 15 章　处理线程和服务器发送事件
- 第 16 章　构建离线的 Web 应用



第 12 章 HTML 5 中的文件与拖放

在 HTML 5 中，专门提供了一个页面层调用的 API 文件，通过调用这个 API 文件中的对象、方法和接口，可以很方便地访问文件的属性或读取文件内容。另外，在 HTML 5 中，还可以将文件进行拖放，即抓取对象以后拖到另一个位置。任何元素都能够被拖放，常见的拖放元素为图片、文字等。

重点案例效果



12.1 选 择 文 件

在 HTML 5 中，可以创建一个 file 类型的<input>元素实现文件的上传功能。只是在 HTML 5 中，该类型的<input>元素新添加了一个 multiple 属性，如果将属性的值设置为 true，则可以在一个元素中实现多个文件的上传。

12.1.1 案例 1——选择单个文件

在 HTML 5 中，当需要创建一个 file 类型的<input>元素上传文件时，可以定义只选择一个文件。

【例 12.1】通过 file 对象选择单个文件(案例文件：ch12\12.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>文件</title>
</head>
<body>
    <form>
    <h3>请选择文件：</h3>
    </p><input type="file" id="fileload" /></p><!-单个文件进行上传-->
    </form>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 12-1 所示，在其中单击“浏览”按钮，打开“选择要加载的文件”对话框，在其中只能选择一个要加载的文件，如图 12-2 所示。




图 12-1 预览效果

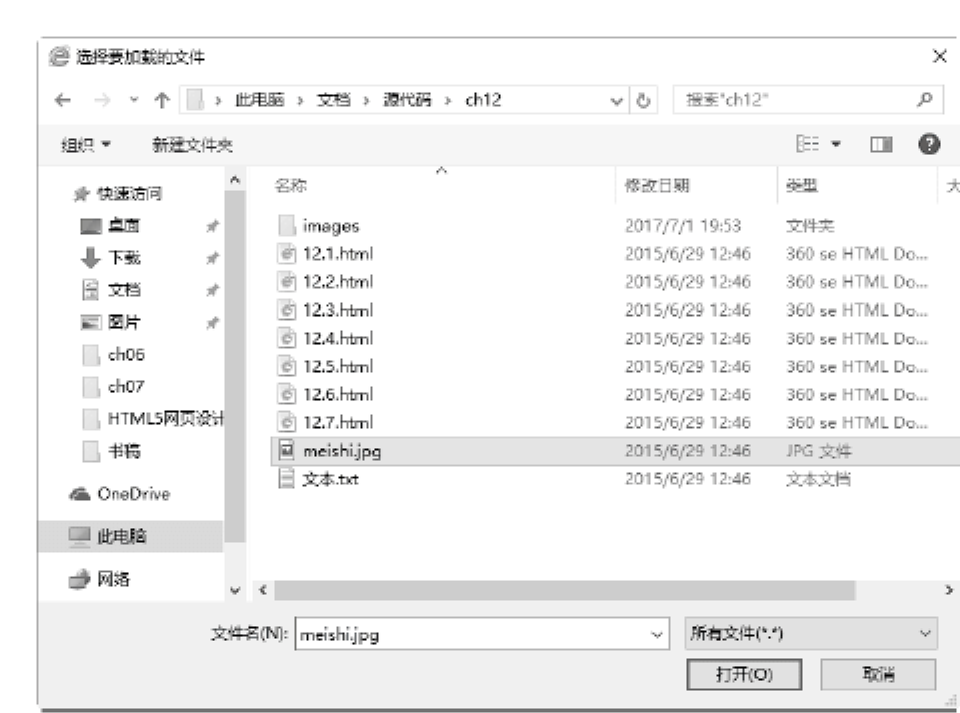


图 12-2 只能选择一个要加载的文件

12.1.2 案例 2——选择多个文件

在 HTML 5 中，除了可以选择单个文件外，还可以通过添加元素的 multiple 属性，实现选择多个文件的功能。



【例 12.2】通过 file 对象选择多个文件(案例文件：ch12\12.2.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<form>
选择文件：<input type="file" multiple="multiple" />
</form>
<p>在浏览文件时可以选取多个文件。</p>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 12-3 所示，在其中单击“浏览”按钮，打开“选择要加载的文件”对话框，在其中可以选择多个要加载的文件，如图 12-4 所示。

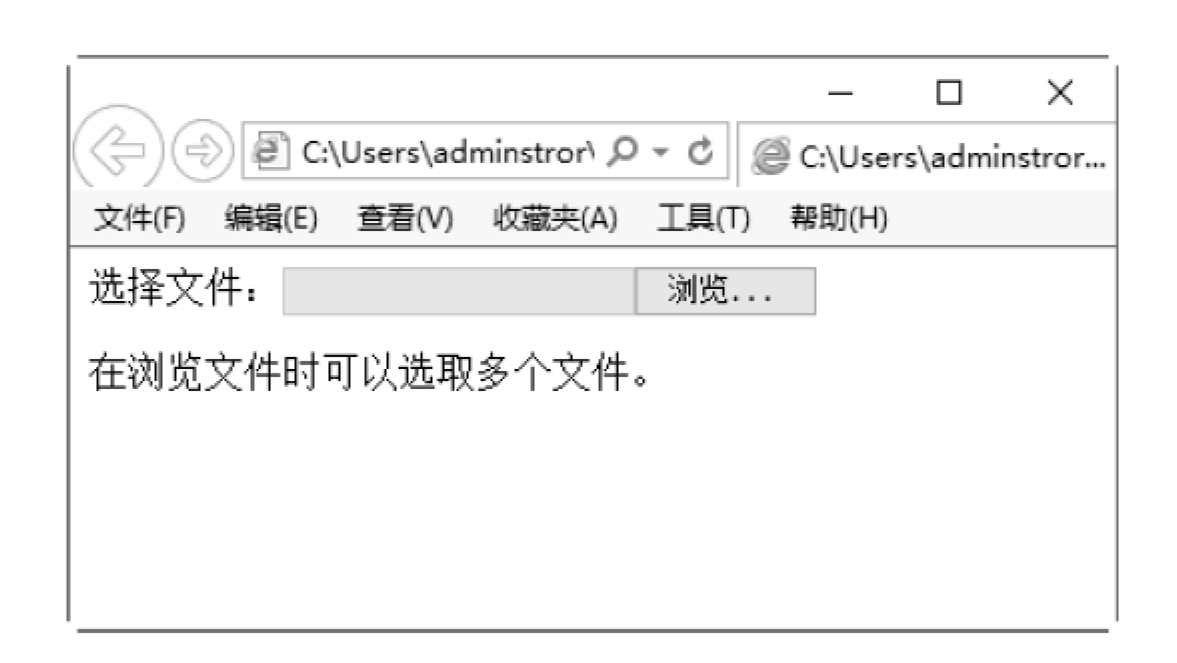


图 12-3 预览效果

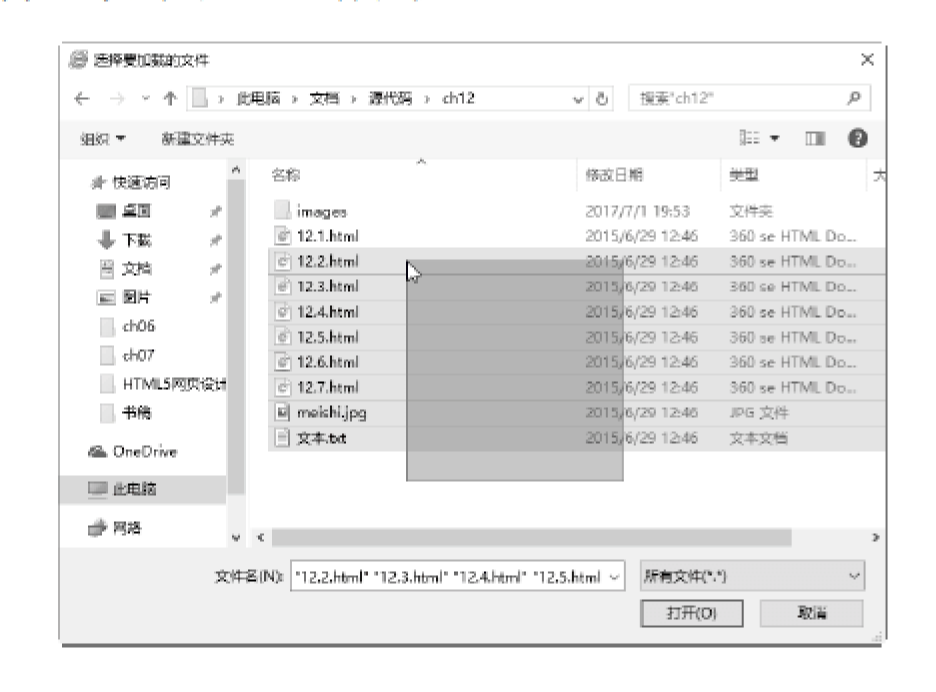


图 12-4 可以选择多个要加载的文件

12.2 使用 FileReader 接口读取文件

使用 Blob 接口可以获取文件的相关信息，如文件名称、大小、类型，但如果想要读取或浏览文件，则需要通过 FileReader 接口。该接口不仅可以读取图片文件，还可以读取文本或二进制文件；同时，根据该接口提供的事件与方法，可以动态侦察文件读取时的详细状态。

12.2.1 检测浏览器是否支持 FileReader 接口

FileReader 接口主要用来把文件读入到内存，并且读取文件中的数据。FileReader 接口提供了一个异步 API，使用该 API 可以在浏览器主线程中异步访问文件系统，读取文件中的数据。到目前为止，并不是所有浏览器都实现了 FileReader 接口。这里提供一种方法可以检查您的浏览器是否对 FileReader 接口提供支持。具体的代码如下:

```
if(typeof FileReader == 'undefined'){
    result.InnerHTML="<p>你的浏览器不支持 FileReader 接口！</p>";
    //使选择控件不可操作
    file.setAttribute("disabled","disabled");
}
```

12.2.2 FileReader 接口的方法

FileReader 接口有 4 个方法，其中 3 个用来读取文件，另外 1 个用来中断读取。无论读取成功或失败，方法并不会返回读取结果，这一结果存储在 result 属性中。FileReader 接口的方法及其描述如表 12-1 所示。

表 12-1 FileReader 接口的方法及其描述

方法名	参数	描述
readAsText	File，[encoding]	将文件以文本方式读取，读取的结果即是这个文本文件中的内容
readAsBinaryString	File	这个方法将文件读取为二进制字符串，通常我们将它送到后端，后端可以通过这段字符串存储文件
readAsDataUrl	File	该方法将文件读取为一串 Data Url 字符串，该方法事实上是将小文件以一种特殊格式的 URL 地址形式直接读入页面。这里的小文件通常是指图像与 html 等格式的文件
abort	(none)	终端读取操作

12.2.3 案例 3——使用 readAsDataURL 方法预览图片

通过 fileReader 接口中的 readAsDataURL()方法，可以获取 API 异步读取的文件数据，另存为数据 URL，将该 URL 绑定<img>元素的 src 属性值，就可以实现图片文件预览的效果。如果读取的不是图片文件，将给出相应的提示信息。

【例 12.3】使用 readAsDataURL 方法预览图片(案例文件：ch12\12.3.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>使用 readAsDataURL 方法预览图片</title>
</head>
<body>
<script type="text/javascript">
var result=document.getElementById("result");
var file=document.getElementById("file");

//判断浏览器是否支持 FileReader 接口
if(typeof FileReader == 'undefined'){
    result.InnerHTML="<p>你的浏览器不支持 FileReader 接口！</p>";
    //使选择控件不可操作
    file.setAttribute("disabled","disabled");
}

function readAsDataURL(){
```

```
    //检验是否为图像文件
    var file = document.getElementById("file").files[0];
    if(!/image\/\w+/.test(file.type)){
        alert("这个不是图片文件，请重新选择！");
        return false;
    }
    var reader = new FileReader();
    //将文件以 Data URL 形式读入页面
    reader.readAsDataURL(file);
    reader.onload=function(e){
        var result=document.getElementById("result");
        //显示文件
        result.innerHTML='<img src="' + this.result +'" alt="" />';
    }
}
</script>
<p>
    <label>请选择一个文件：</label>
    <input type="file" id="file" />
    <input type="button" value="读取图像" onclick="readAsDataURL()" />
</p>
<div id="result" name="result"></div>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 12-5 所示。在页面中单击“浏览”按钮，打开“选择要加载的文件”对话框，在其中选择需要预览的图片文件，如图 12-6 所示。



图 12-5 预览效果

图 12-6 “选择要加载的文件”对话框

选择完毕后，在“选择要加载的文件”对话框中单击“打开”按钮，返回到 IE 11.0 中，然后单击“读取图像”按钮，即可在页面的下方显示添加的图片，如图 12-7 所示。

如果在“选择要加载的文件”对话框中选择的不是图片文件，当在 IE 11.0 中单击“读取图像”按钮后，就会给出相应的提示信息，如图 12-8 所示。



图 12-7　显示图片



图 12-8　信息提示框

12.2.4　案例 4——使用 readAsText 方法读取文本文件

使用 FileReader 接口中的 readAsTextO 方法，可以将文件以文本编码的方式进行读取，即可以读取上传文本文件的内容；其实现的方法与读取图片基本相似，只是读取文件的方式不一样。

【例 12.4】使用 readAsText 方法读取文本文件(案例文件：ch12\12.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>使用 readAsText 方法读取文本文件</title>
</head>
<body>
<script type="text/javascript">
var result=document.getElementById("result");
var file=document.getElementById("file");

//判断浏览器是否支持 FileReader 接口
if(typeof FileReader == 'undefined'){
    result.InnerHTML="<p>你的浏览器不支持 FileReader 接口! </p>";
    //使选择控件不可操作
    file.setAttribute("disabled","disabled");
}
function readAsText(){
    var file = document.getElementById("file").files[0];
    var reader = new FileReader();
    //将文件以文本形式读入页面
    reader.readAsText(file,"gb2312");
    reader.onload=function(f){
        var result=document.getElementById("result");
        //显示文件
        result.innerHTML=this.result;
    }
```

```
}
</script>
<p>
    <label>请选择一个文件：</label>
    <input type="file" id="file" />
    <input type="button" value="读取文本文件" onclick="readAsText()" />
</p>
<div id="result" name="result"></div>
</body>
</html>
```

在 IE 11.0 中预览，效果如图 12-9 所示。在页面中单击“浏览”按钮，打开“选择要加载的文件”对话框，在其中选择需要读取的文件，如图 12-10 所示。

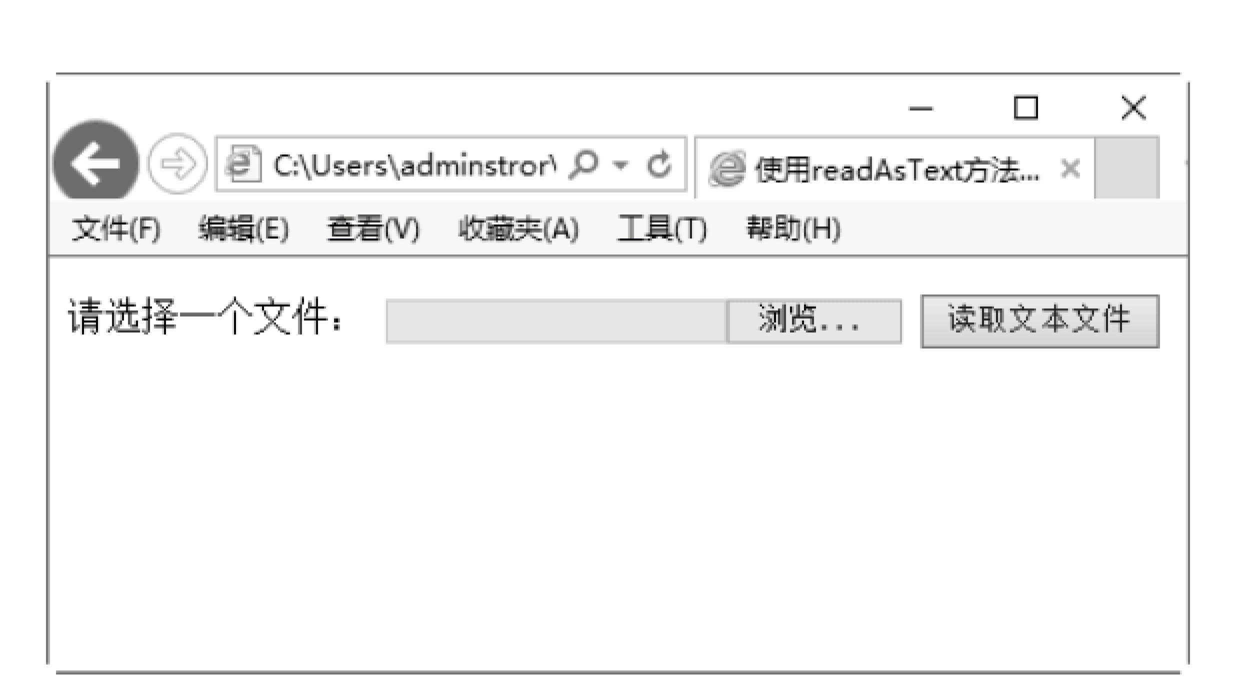


图 12-9　预览效果

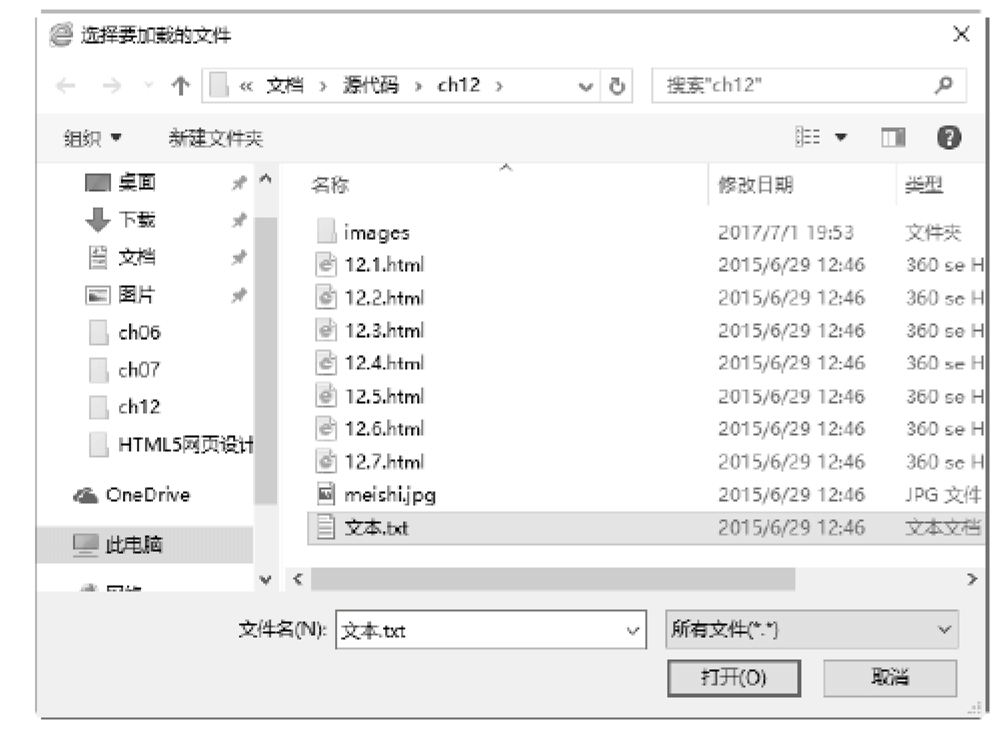


图 12-10　选择要读取的文件

选择完毕后，在“选择要加载的文件”对话框中单击“打开”按钮，返回到 IE 11.0 中，然后单击“读取文本文件”按钮，即可在页面的下方读取文本文件中的信息，如图 12-11 所示。

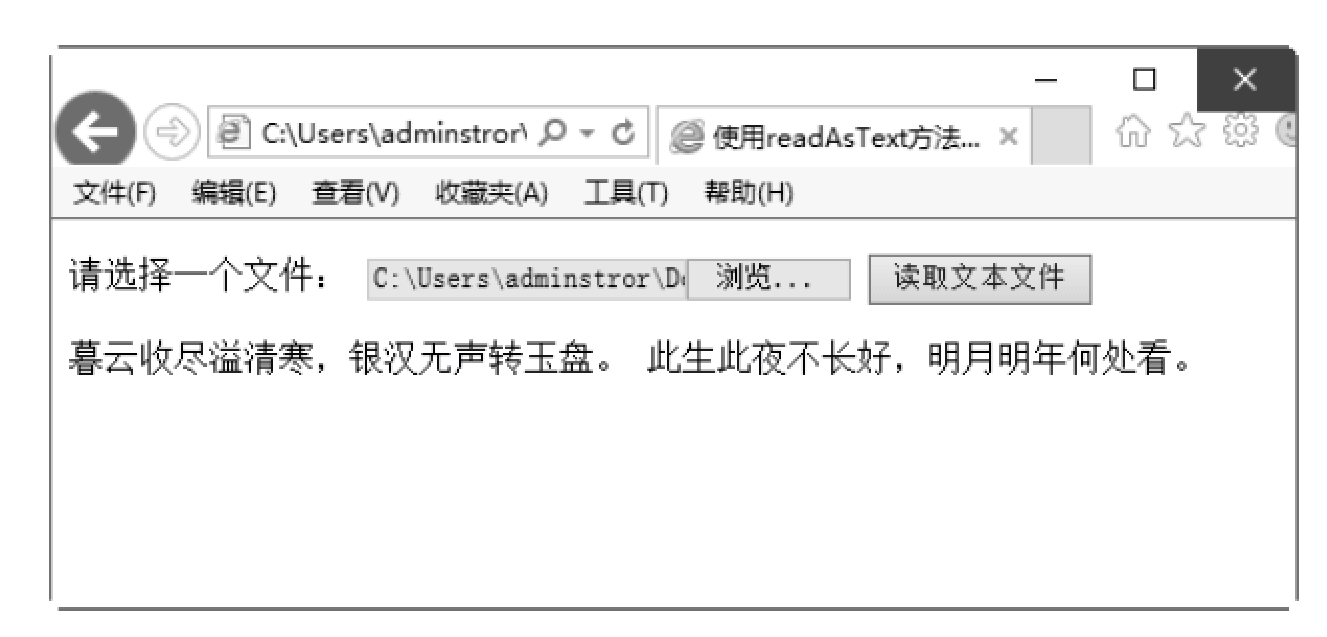


图 12-11　读取文本信息

12.3　使用 HTML 5 实现文件的拖放

使用 HTML 5 可以实现拖放效果。常用的实现方法是利用 HTML 5 新增加的事件 drag 和 drop。

12.3.1 认识文件拖放的过程

在 HTML 5 中实现文件的拖放主要有以下 4 个步骤。

1. 设置元素为可拖放

首先，为了使元素可拖动，把 draggable 属性设置为 true，具体代码如下：

```
<img draggable="true" />
```

2. 拖动什么

实现拖放的第二步就是设置拖动的元素，常见的元素有图片、文字、动画等。实现拖放功能的是 ondragstart 和 setData()，即规定当元素被拖动时，会发生什么。

例如：在上面的例子中，ondragstart 属性调用了一个函数 drag(event)，它规定了被拖动的数据。

dataTransfer.setData()方法设置被拖数据的数据类型和值，具体代码如下：

```
function drag(ev)
{
ev.dataTransfer.setData("Text",ev.target.id);
}
```

在这个例子中，数据类型是"Text"，值是可拖动元素的 id ("drag1")。

3. 放到何处

实现拖放功能的第三步就是将可拖放元素放到何处，实现该功能的事件是 ondragover，在默认情况下，无法将数据/元素放置到其他元素中。如果需要设置允许放置，用户必须阻止对元素的默认处理方式。

这就需要通过调用 ondragover 事件的 event.preventDefault() 方法，具体代码如下：

```
event.preventDefault()
```

4. 进行放置

当放置被拖数据时，就会发生 drop 事件。在上面的例子中，ondrop 属性调用了一个函数 drop(event)，具体代码如下：

```
function drop(ev)
{
ev.preventDefault();
var data=ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(data));
}
```

12.3.2 浏览器支持情况

不用的浏览器版本对拖放技术的支持情况是不同的，如表 12-2 所示是常见浏览器对拖放

技术的支持情况。

表 12-2　浏览器对拖放技术的支持情况

浏览器名称	支持 Web 存储技术的版本
Internet Explorer	Internet Explorer 9 及更高版本
Firefox	Firefox 3.6 及更高版本
Opera	Opera 12.0 及更高版本
Safari	Safari 5 及更高版本
Chrome	Chrome 5 及更高版本

12.3.3　案例 5——在网页中拖放图片

下面给出一个简单的拖放实例，该实例主要实现的功能就是把一张图片拖放到一个矩形中。

【例 12.5】将图片拖放至矩形中(案例文件：ch12\12.5.html)。

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
#div1 {width:150px;height:150px;padding:10px;border:1px solid #aaaaaa;}
</style>
<script type="text/javascript">
function allowDrop(ev)
{
ev.preventDefault();
}
function drag(ev)
{
ev.dataTransfer.setData("Text",ev.target.id);
}
function drop(ev)
{
ev.preventDefault();
var data=ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>
<p>请把图片拖放到矩形中：</p>
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
<br />
<img id="drag1" src="images/2.gif" draggable="true" ondragstart="drag(event)"
/>
</body>
</html>
```

将上述代码保存为.html 格式，在 IE 11.0 中预览，效果如图 12-12 所示。

可以看到当选中图片后，在不释放鼠标的情况下，可以将其拖放到矩形框中，如图 12-13 所示。



图 12-12　预览效果

图 12-13　拖放图片

代码解释如下。

(1) 调用 preventDefault()来避免浏览器对数据的默认处理(drop 事件的默认行为是以链接形式打开)。

(2) 通过 dataTransfer.getData("Text")方法获得被拖的数据。该方法将返回在 setData()方法中设置为相同类型的任何数据。

(3) 被拖数据是被拖元素的 id ("drag1")。

(4) 把被拖元素追加到放置元素(目标元素)中。

12.4　综合案例——在网页中来回拖放图片

下面再给出一个具体实例，该实例所实现的效果就是在网页中来回拖放图片。具体代码如下：

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
#div1, #div2
{float:left; width:100px; height:35px; margin:10px;padding:10px;border:1px
solid #aaaaaa;}
</style>
<script type="text/javascript">
function allowDrop(ev)
{
ev.preventDefault();
}
function drag(ev)
{
```

```
ev.dataTransfer.setData("Text",ev.target.id);
}
function drop(ev)
{
ev.preventDefault();
var data=ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">
  <img src="images/1.gif" draggable="true" ondragstart="drag(event)"
id="drag1" />
</div>
<div id="div2" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
</body>
</html>
```

在记事本中输入这些代码，然后将其保存为.html 格式。使用 IE 11.0 打开文件，即可在页面中查看效果。选中网页中的图片，即可在两个矩形中来回拖放，如图 12-14 所示。

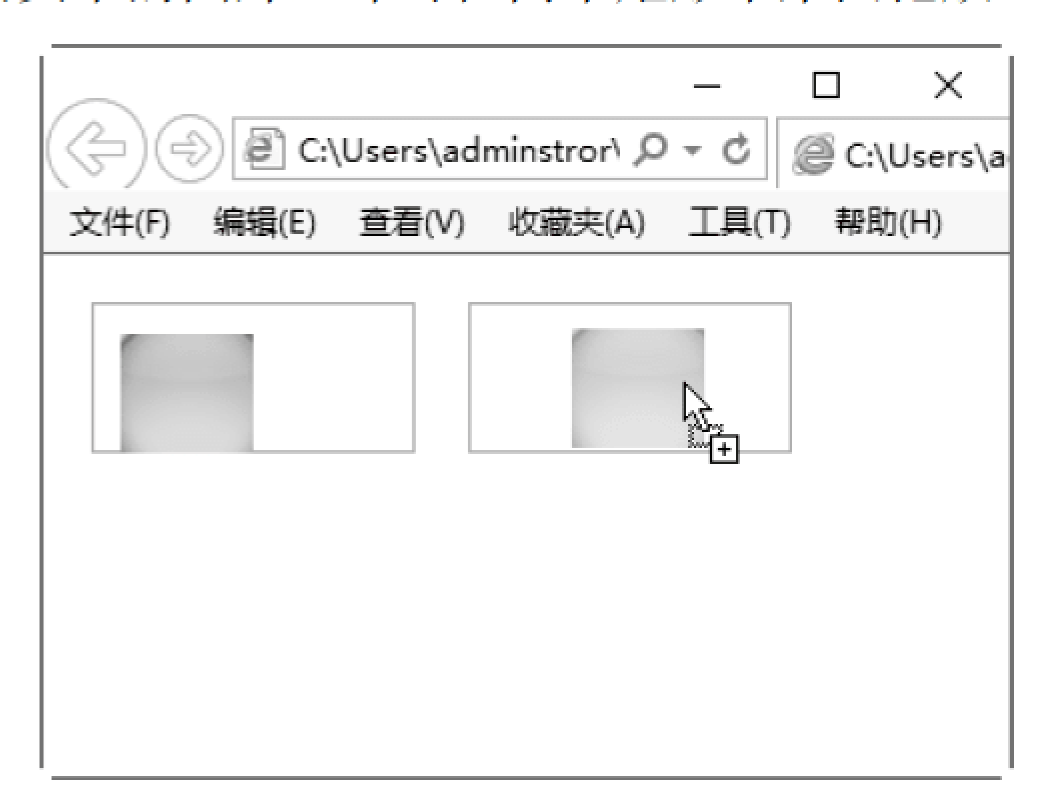


图 12-14　预览效果

12.5　跟我学上机——在网页中拖放文字

在了解了 HTML 5 的拖放技术后，下面给出一个具体实例，该实例所实现的效果就是在网页中拖放文字。

【例 12.6】在网页中拖放文字(案例文件：ch12\12.6.html)。

```
<!DOCTYPE HTML>
<html>
<head>
<title>拖放文字</title>
<style>
body {
    font-family: 'Microsoft YaHei';
}
```

```
div.drag {
    background-color:#AACCFF;
    border:1px solid #666666;
    cursor:move;
    height:100px;
    width:100px;
    margin:10px;
    float:left;
}
div.drop {
    background-color:#EEEEEE;
    border:1px solid #666666;
    cursor: pointer;
    height:150px;
    width:150px;
    margin:10px;
    float:left;
}
</style>
</head>
<body>
<div draggable="true" class="drag"
     ondragstart="dragStartHandler(event)">拖拽我吧!</div>
<div class="drop"
     ondragenter="dragEnterHandler(event)"
     ondragover="dragOverHandler(event)"
     ondrop="dropHandler(event)">放这里吧!<ol /></div>
<script>
var internalDNDType = 'text';
function dragStartHandler(event) {
   event.dataTransfer.setData(internalDNDType,
                              event.target.textContent);
    event.effectAllowed = 'move';
}
// dragEnter 事件
function dragEnterHandler(event) {
   if (event.dataTransfer.types.contains(internalDNDType))
      if (event.preventDefault) event.preventDefault();}
// dragOver 事件
function dragOverHandler(event) {
    event.dataTransfer.dropEffect = 'copy';
    if (event.preventDefault) event.preventDefault();
}
function dropHandler(event) {
    var data = event.dataTransfer.getData(internalDNDType);
    var li = document.createElement('li');
    li.textContent = data;
    event.target.lastChild.appendChild(li);
}
</script>
</body>
</html>
```

下面介绍实现拖放的具体操作步骤。

step 01 将上述代码保存为.html 格式的文件，在 IE 11.0 中预览，效果如图 12-15 所示。

step 02 选中左边矩形中的元素，将其拖曳到右边的方框中，如图 12-16 所示。

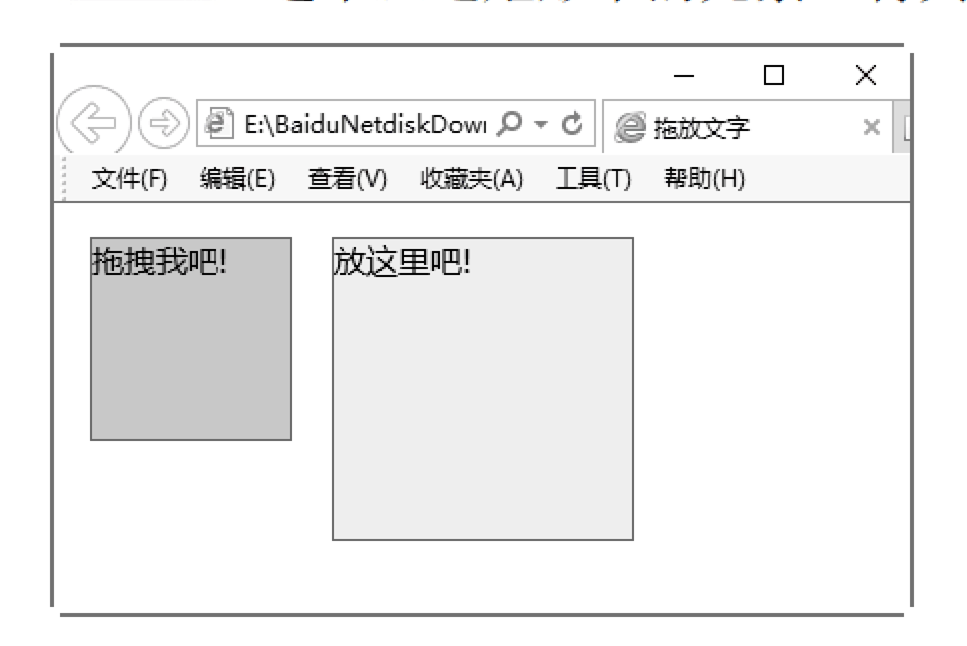


图 12-15 预览效果

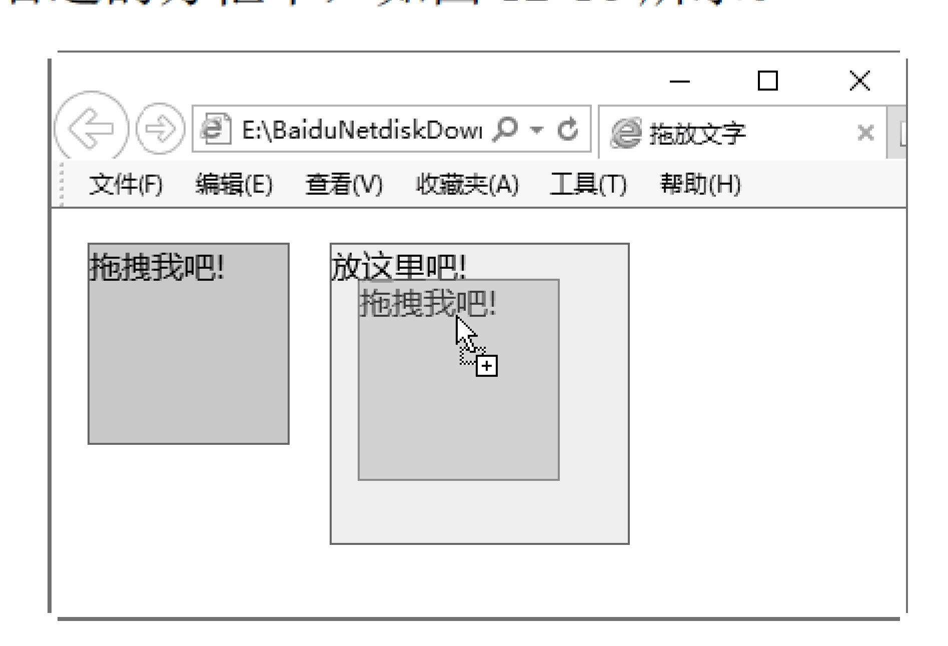


图 12-16 选中被拖放文字

step 03 释放鼠标，可以看到拖放之后的效果，如图 12-17 所示。

step 04 还可以多次拖放文字元素，效果如图 12-18 所示。

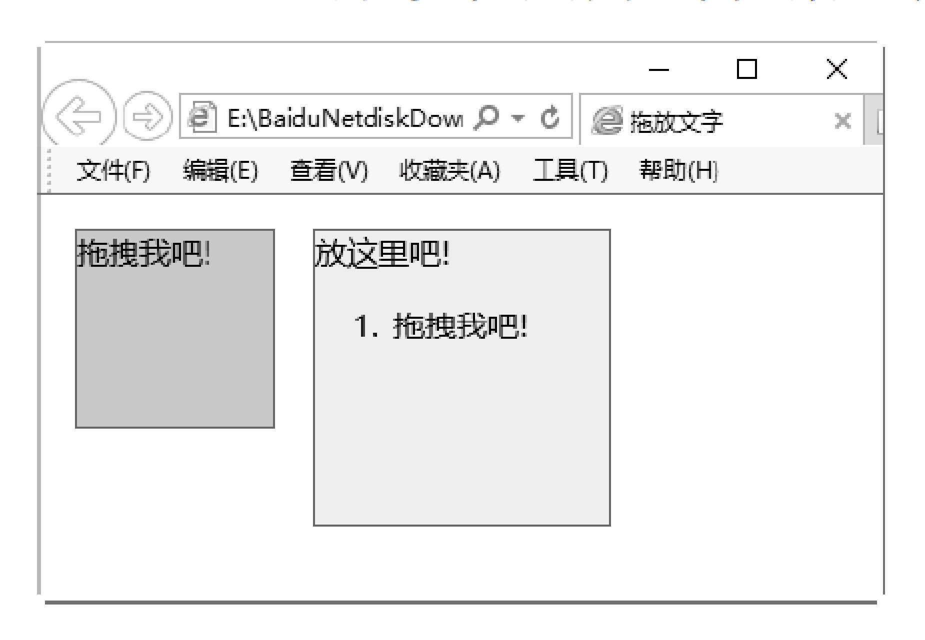


图 12-17 拖放一次

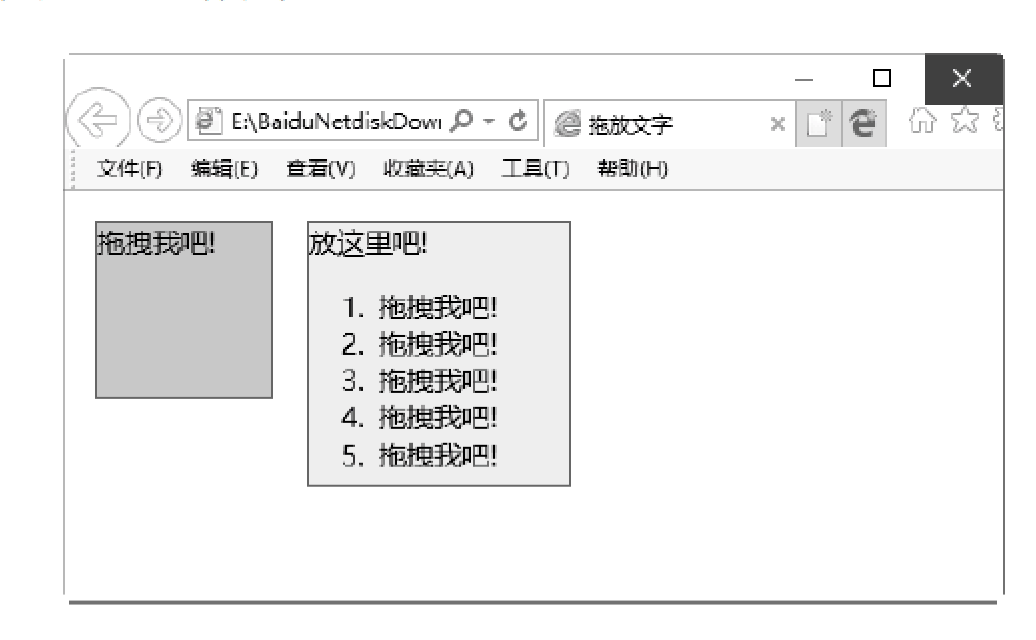


图 12-18 拖放多次

12.6 高手解惑

疑问 1：在 HTML 5 中，实现拖放效果的方法是唯一的吗？

答：在 HTML 5 中，实现拖放效果的方法并不是唯一的。除了可以使用事件 drag 和 drop 外，还可以利用 canvas 标签来实现。

疑问 2：在 HTML 5 中，可拖放的对象只有文字和图像吗？

答：在默认情况下，图像、链接和文本是可以拖动的，也就是说，不用额外编写代码，用户就可以拖动它们。文本只有在被选中的情况下才能拖动，而图像和链接在任何时候都可以拖动。

如果让其他元素可以拖动也是可能的。HTML 5 为所有 HTML 元素规定了一个 draggable 属性，表示元素是否可以拖动。图像和链接的 draggable 属性自动被设置成了 true，而其他元素这个属性的默认值都是 false。要想让其他元素可以拖动，或者让图像或链接不能拖动，都可以设置这个属性。

疑问 3：在 HTML 5 中，读取记事本文件中的中文内容时显示乱码怎么办呢？

答：读者需要特别注意的是，读取文件内容时可能显示乱码，如图 12-19 所示。

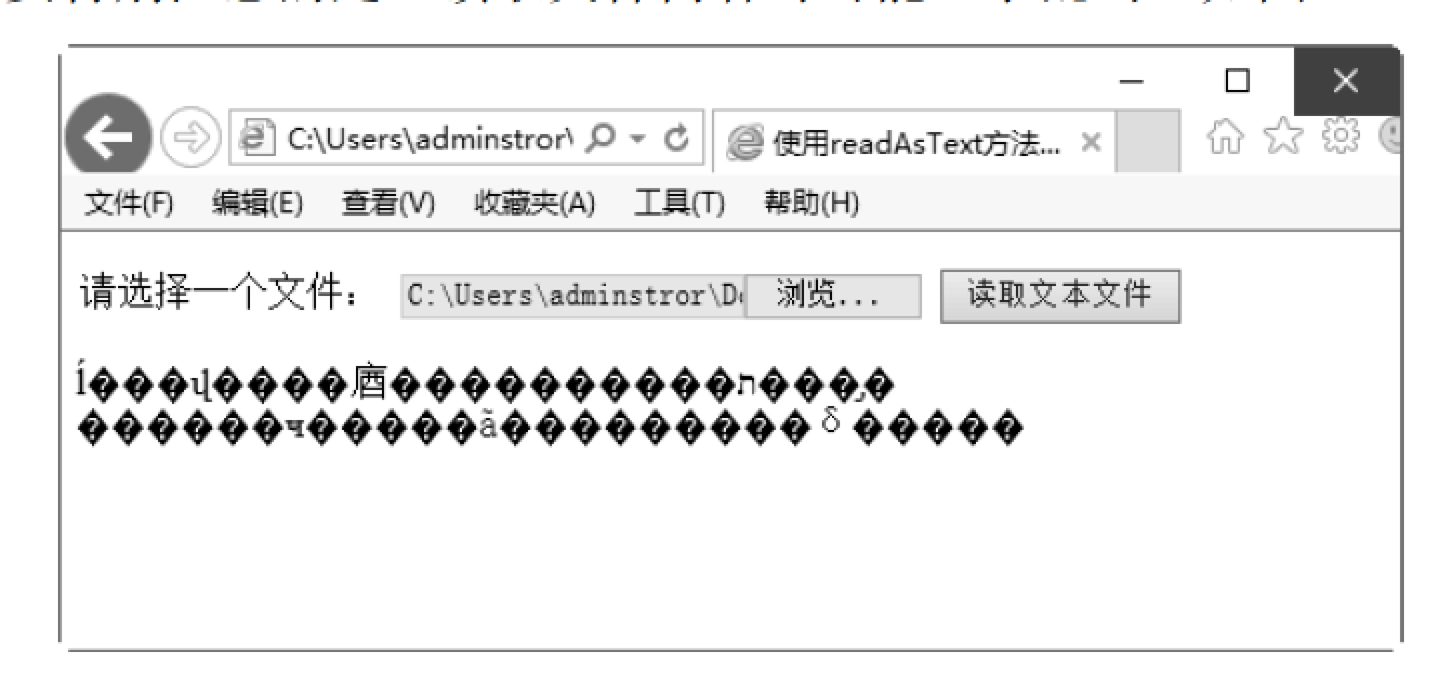


图 12-19　读取文件内容时显示乱码

这里的原因是在读取文件时，没有设置读取的编码方式。例如下面的代码：

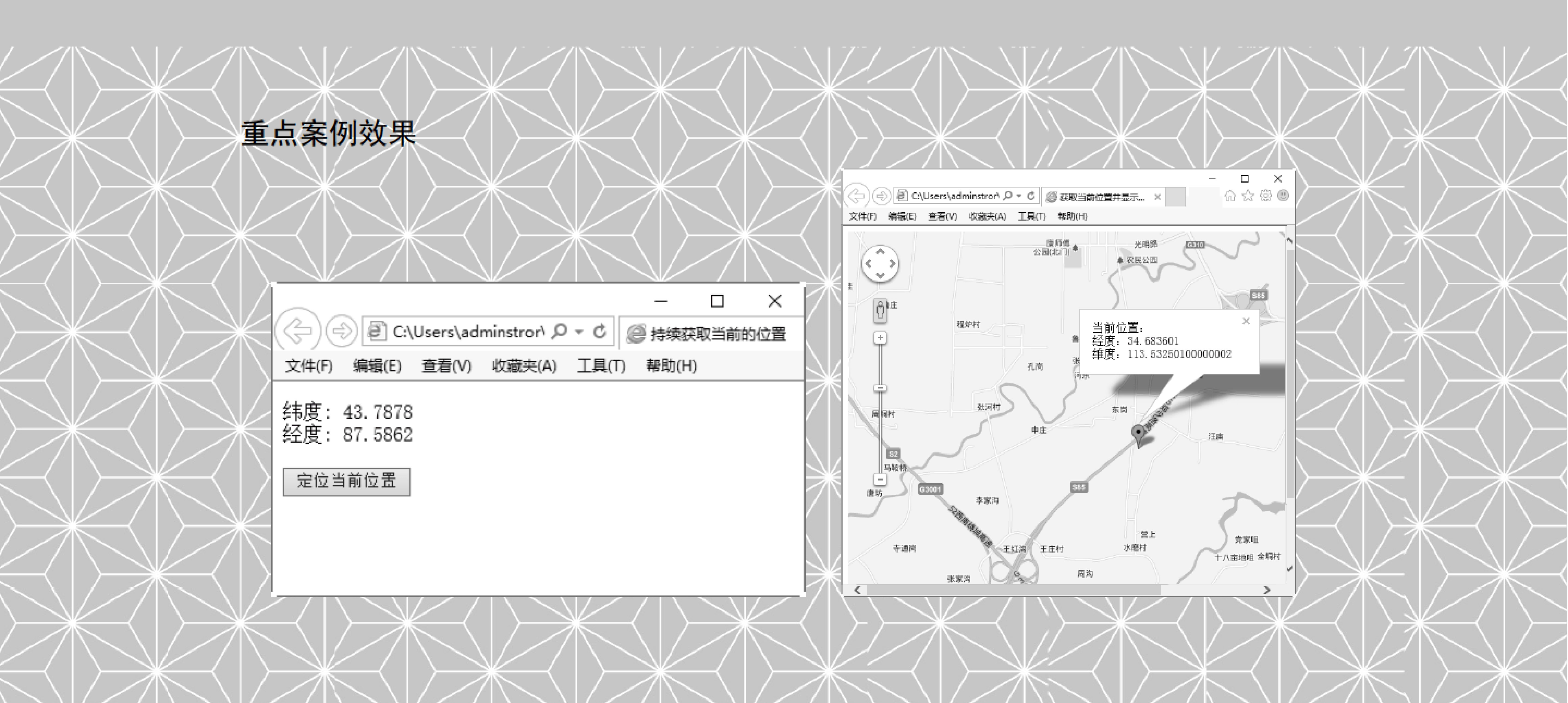
```
reader.readAsText(file);
```

设置读取的格式，如果是中文内容，修改如下：

```
reader.readAsText(file,"gb2312");
```

第 13 章 定位地理位置技术

根据访问者访问网站的方式，有多种获取地理位置的方法。本章主要介绍如何利用 Geolocation API 来获取地理位置。



13.1 Geolocation API 获取地理位置

在 HTML 5 网页代码中，通过一些有用的 API，可以查找访问者当前的位置。

13.1.1 地理地位的原理

由于访问者浏览网站的方式不同，可以通过下列方式确定其位置。

(1) 如果网站浏览者使用电脑上网，通过获取浏览者的 IP 地址，从而确定其具体位置。

(2) 如果网站浏览者通过手机上网，通过获取浏览者的手机信号接收塔，从而确定其具体位置。

(3) 如果网站浏览者的设备上具有 GPS 硬件，通过获取 GPS 发出的载波信号，可以获取其具体位置。

(4) 如果网站浏览者通过无线上网，可以通过无线网络连接获取其具体位置。



API 是应用程序的编程接口，是一些预先定义的函数，目的是让开发人员调用需要的功能程序，而又无须访问源码，或理解内部工作机制的细节。

13.1.2 获取定位信息的方法

在了解了地理定位的原理后，下面介绍获取定位信息的方法，根据访问者访问网站的方式，可以通过下列方法之一确定地理位置。

(1) 利用 IP 地址定位。

(2) 利用 GPS 功能定位。

(3) 利用 Wi-Fi 定位。

(4) 利用 Wi-Fi 和 GPRS 联合定位。

(5) 利用用户自定义定位数据定位。

使用上述的哪种方法将取决于浏览器和设备的功能。然后，浏览器确定位置并将其传输回地理位置，但需要注意的是无法保证返回的位置是设备的实际地理位置。因为，这涉及一个隐私问题，并不是每个人都想与您共享他的位置。

13.1.3 常用地理定位方法

通过地理定位，可以确定用户的当前位置，并能获取用户地理位置的变化情况。其中，最常用的就是 API 中的 getCurrentposition 方法。

getCurrentposition 方法的语法格式如下：

```
void getCurrentPosition(successCallback, errorCallback, options);
```

其中，successCallback 参数是指在位置成功获取时用户想要调用的函数名称；errorCallback 参数是指在位置获取失败时用户想要调用的函数名称；options 参数指出地理

定位时的属性设置。



访问用户位置是耗时的操作，同时属于隐私问题，还要取得用户的同意。

如果地理定位成功，新的 Position 对象将调用 displayOnMap 函数，显示设备的当前位置。

那么 Position 对象的含义是什么呢？作为地理定位的 API，Position 对象包含位置确定时的时间戳(timestamp)和包含位置的坐标(coords)，具体语法格式如下:

```
Interface position
 {
readonly attribute Coordinates cords;
readonly attribute DOMTimeStamp timestamp;
} ;
```

13.1.4 判断浏览器是否支持 HTML 5 获取地理位置信息

在用户试图使用地理定位之前，应该先确保浏览器是否支持 HTML 5 获取地理位置信息。这里介绍判断的方法。具体代码如下：

```
function init()
if (navigator.geolocation) {
//获取当前地理位置信息
navigator.geolocation.getCurrentPosition(onSuccess, onError, options);
} else {
alert("你的浏览器不支持 HTML 5 来获取地理位置信息。");
}
```

该代码解释如下。

1. onSuccess

该函数是获取当前位置信息成功时执行的回调函数。

在 onSuccess 回调函数中，用到了参数 position，代表一个具体的 position 对象，表示当前位置。其具有如下属性。

(1) latitude：当前地理位置的纬度。

(2) longitude：当前地理位置的经度。

(3) altitude：当前位置的海拔高度(不能获取时为 null)。

(4) accuracy：获取到的纬度和经度的精度(以米为单位)。

(5) altitudeAccuracy：获取到的海拔高度的精度(以米为单位)。

(6) heading：设备的前进方向。用面朝正北方向的顺时针旋转角度来表示(不能获取时为 null)。

(7) speed：设备的前进速度(以米/秒为单位，不能获取时为 null)。

(8) timestamp：获取地理位置信息时的时间。

2. onError

该函数是获取当前位置信息失败时所执行的回调函数。

在 onError 回调函数中，用到了 error 参数。其具有如下属性。

(1) code：错误代码，有如下值。

① 用户拒绝了位置服务(属性值为 1)。

② 获取不到位置信息(属性值为 2)。

③ 获取信息超时错误(属性值为 3)。

(2) message：字符串，包含了具体的错误信息。

3. options

options 是一些可选熟悉列表。在 options 参数中，可选属性如下。

(1) enableHighAccuracy：是否要求高精度的地理位置信息。

(2) timeout：设置超时时间(单位为毫秒)。

(3) maximumAge：对地理位置信息进行缓存的有效时间(单位为毫秒)。

13.1.5 指定纬度和经度坐标

对于地理定位成功后，将调用 displayOnMap 函数。此函数如下：

```
function displayOnMap(position)
{
var latitude=positon.coords.latitude;
var longitude=postion.coords.longitude;
}
```

其中第一行函数从 Position 对象获取 coordinates 对象，主要由 API 传递给程序调用。第三行和第四行中定义了两个变量，latitude 和 longitude 属性存储在定义的两个变量中。

为了在地图上显示用户的具体位置，可以利用地图网站的 API。下面以使用百度地图为例进行讲解，则需要使用 Baidu Maps JavaScript API。在使用此 API 前，需要在 HTML 5 页面中添加一个引用，具体代码如下：

```
<--baidu maps API>
<script type= "text/javascript" scr= "http://api.map.baidu.com/api?key=*&v=
1.0&services=true">
</script>
```

其中“*”代码注册到 key。注册 key 的方法为：在 http：//openapi.baidu.com/map/index.html 网页中，注册百度地图 API，然后输入需要内置百度地图页面的 URL 地址，生成 API 密钥，然后将 key 文件复制保存。

虽然已经包含了 Baidu Maps JavaScript，但是页面中还不能显示内置的百度地图，还需要添加 html 语言，让地图从程序转化为对象。需要加入以下源代码：

```
<script
type="text/javascript"scr="http://api.map.baidu.com/api?key=*&v=1.0&service
s=true">
```

```
</script>
<div style="width:600px;height:220px;border:1px solid gary;margin-
top:15px;" id="container">
</div>
<script type="text/javascript">
var map = new BMap.Map("container");
map.centerAndZoom(new BMap.Point(***,***),17);
map.addControl(new BMap.NavigationControl());
map.addControl(new BMap.ScaleControl());
map.addControl(new BMap.OverviewMapControl());
var local = new BMap.LocalSearch(map,
{
enderOptions:{map: map}
}
);
local.search("输入搜索地址");
</script>
```

上述代码分析如下。

(1) 其中前 2 行主要是把 baidu map API 程序植入源码中。

(2) 第 3 行在页面中设置一个标签，包括宽度和长度，用户可以自己调整；border=1px 是定义外框的宽度为 1 个像素，solid 为实线，gray 为边框显示颜色，margin-top 为该标签与上部的距离。

(3) 第 7 行为地图中自己位置的坐标。

(4) 第 8 到第 10 行为植入地图缩放控制工具。

(5) 第 11 到第 16 行为地图中自己的位置，只需要在 local search 后填入自己的位置名称即可。

13.1.6　获取当前位置的经度与纬度

如下代码为使用纬度和经度定位坐标的案例。

step 01 打开记事本文件，在其中输入如下代码：

```
<!DOCTYPE html>
<html>
<head>
<title>纬度和经度坐标</title>
<style>
body {background-color:#fff;}
</style>
</head>
<body>
<p id="geo loc"><p>
<script>
function getElem(id) {
    return typeof id === 'string' ? document.getElementById(id) : id;
 }

 function show it(lat, lon) {
    var str = '您当前的位置，纬度：' + lat + '，经度：' + lon;
```

```
    getElem('geo loc').innerHTML = str;
 }
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function(position) {
       show it(position.coords.latitude, position.coords.longitude);
    },
    function(err) {
       getElem('geo_loc').innerHTML = err.code + "|" + err.message;
    });
 } else {
    getElem('geo loc').innerHTML = "您当前使用的浏览器不支持 Geolocation 服务";
 }
</script>
</body>
</html>
```

step 02 使用 IE 11.0 打开网页文件，由于使用 HTML 5 定位功能首先要由用户允许网页运行脚本，所以弹出如图 13-1 所示的提示框，单击“允许阻止的内容”按钮。

step 03 弹出想要跟踪实际位置的提示信息，单击“允许一次”按钮，如图 13-2 所示。



图 13-1　允许网页运行脚本

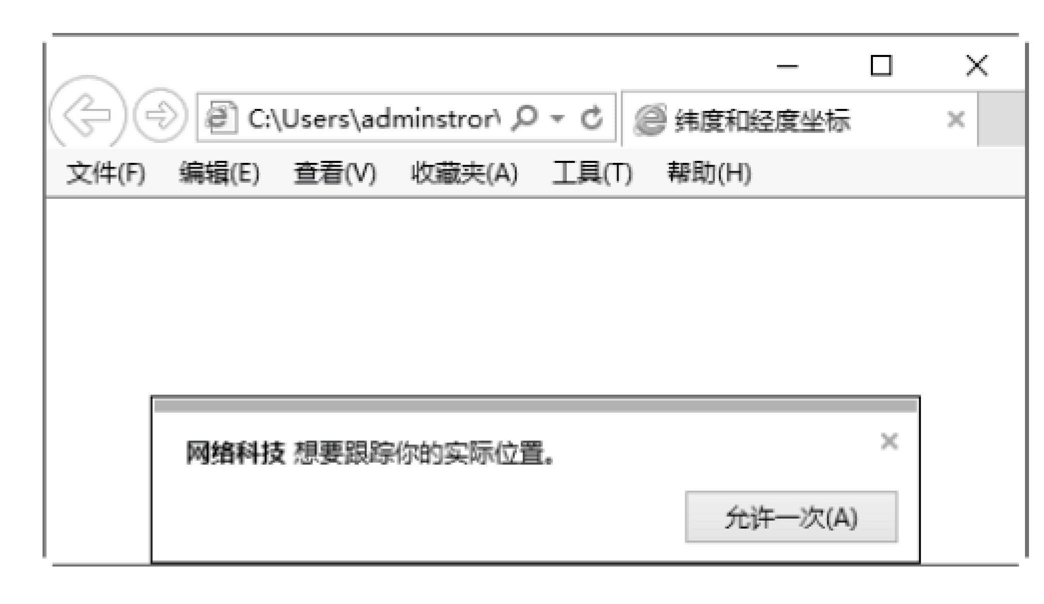


图 13-2　允许跟踪实际位置

step 04 在页面中显示了当前页面打开时所处的地理位置，其位置为使用者的 IP 或 GPS 定位地址，如图 13-3 所示。

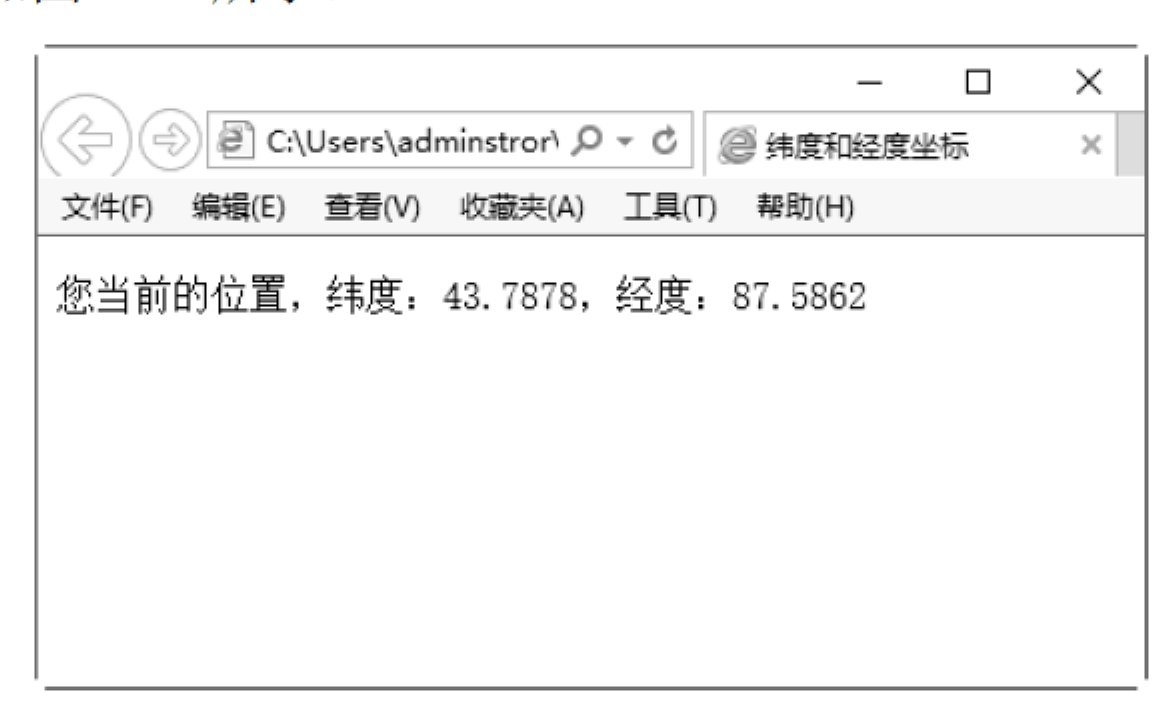


图 13-3　显示的地理位置



每次使用浏览器打开网页时都会提醒是否允许跟踪实际位置，为了安全，用户应当妥善使用地址共享功能。



13.1.7 处理错误和拒绝

getCurrentPosition() 方法的第二个参数用于处理错误。它规定当获取用户位置失败时运行的函数。例如以下代码：

```
function showError(error)
{
    switch(error.code)
    {
        case error.PERMISSION_DENIED:
            x.innerHTML="用户拒绝对获取地理位置的请求。"
            break;
        case error.POSITION_UNAVAILABLE:
            x.innerHTML="位置信息是不可用的。"
            break;
        case error.TIMEOUT:
            x.innerHTML="请求用户地理位置超时。"
            break;
        case error.UNKNOWN_ERROR:
            x.innerHTML="未知错误。"
            break;
    }
}
```

其中 PERMISSION_DENIED 表示用户不允许地理定位；POSITION_UNAVAILABLE 表示无法获取当前位置；TIMEOUT 表示操作超时；UNKNOWN_ERROR 表示未知的错误。针对不同的错误类型，将弹出不同的错误信息。

13.2 目前浏览器对地理定位的支持情况

不同的浏览器版本对地理定位技术的支持情况是不同的。表 13-1 是常见浏览器对地理定位的支持情况。

表 13-1 常见浏览器对地理定位的支持情况

浏览器名称	支持 Web 存储技术的版本
Internet Explorer	Internet Explorer 9 及更高版本
Firefox	Firefox 3.5 及更高版本
Opera	Opera 10.6 及更高版本
Safari	Safari 5 及更高版本
Chrome	Chrome 5 及更高版本
Android	Android 2.1 及更高版本

13.3 综合案例——在网页中调用 Google 地图

本实例介绍如何在网页中调用 Google 地图，以获取当前设备物理地址的经度与纬度。具体操作步骤如下。

step 01 调用 Google Map，代码如下：

```
<!DOCTYPE html>
<head>
<title>获取当前位置并显示在 google 地图上</title>
<script type="text/javascript" src="http://maps.google.com/maps/api/
js?sensor=false"></script>
<script type="text/javascript">
```

step 02 获取当前地理位置，代码如下：

```
navigator.geolocation.getCurrentPosition(function (position) {
var coords = position.coords;
//console.log(position);
```

step 03 设定地图参数，代码如下：

```
var latlng = new google.maps.LatLng(coords.latitude, coords.longitude);
var myOptions = {
zoom: 14, //设定放大倍数
center: latlng, //将地图中心点设定为指定的坐标点
mapTypeId: google.maps.MapTypeId.ROADMAP //指定地图类型
};
```

step 04 创建地图，并在页面中显示，代码如下：

```
var map = new google.maps.Map(document.getElementById("map"), myOptions);
```

step 05 在地图上创建标记，代码如下：

```
var marker = new google.maps.Marker({
position: latlng, //将前面设定的坐标标注出来
map: map //将该标注设置在刚才创建的 map 中
});
```

step 06 创建窗体内的提示内容，代码如下：

```
var infoWindow = new google.maps.InfoWindow({
content: "当前位置：<br/>经度：" + latlng.lat() + "<br/>纬度：" + latlng.lng()
//提示窗体内的提示信息
});
```

step 07 打开提示窗口，代码如下：

```
infoWindow.open(map, marker);
},
```

step 08 根据需要再编写其他相关代码，如处理错误的方法和打开地图的大小等。查看此时页面相应的 HTML 源代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>获取当前位置并显示在 google 地图上</title>
<script type="text/javascript" src="http://maps.google.com/maps/api/
js?sensor=false"></script>
<script type="text/javascript">
function init() {
if (navigator.geolocation) {
//获取当前地理位置
navigator.geolocation.getCurrentPosition(function (position) {
var coords = position.coords;
//console.log(position);
//指定一个 google 地图上的坐标点，同时指定该坐标点的横坐标和纵坐标
var latlng = new google.maps.LatLng(coords.latitude, coords.longitude);
var myOptions = {
zoom: 14, //设定放大倍数
center: latlng, //将地图中心点设定为指定的坐标点
mapTypeId: google.maps.MapTypeId.ROADMAP //指定地图类型
};
//创建地图，并在页面 map 中显示
var map = new google.maps.Map(document.getElementById("map"), myOptions);
//在地图上创建标记
var marker = new google.maps.Marker({
position: latlng, //将前面设定的坐标标注出来
map: map //将该标注设置在刚才创建的 map 中
});
//标注提示窗口
var infoWindow = new google.maps.InfoWindow({
content: "当前位置：<br/>经度：" + latlng.lat() + "<br/>纬度：" + latlng.lng()
//提示窗体内的提示信息
});
//打开提示窗口
infoWindow.open(map, marker);
},
function (error) {
//处理错误
switch (error.code) {
case 1:
alert("位置服务被拒绝。");
break;
case 2:
alert("暂时获取不到位置信息。");
break;
case 3:
alert("获取信息超时。");
break;
default:
alert("未知错误。");
break;
}
});
} else {
alert("你的浏览器不支持 HTML 5 来获取地理位置信息。");
```

```
}
}
</script>
</head>
<body onload="init()">
<div id="map" style="width: 800px; height: 600px"></div>
</body>
</html>
```

step 09 保存网页后，即可查看最终效果，如图 13-4 所示。



图 13-4　调用 Google 地图

13.4　跟我学上机——持续获取用户移动后的位置

类似汽车上的 GPS 定位系统一样，在 HTML 5 网页中，用户也可以持续获取到移动设备的位置。这里使用 watchPosition()方法，不仅可以返回用户的当前位置，而且可以继续返回用户移动时的更新位置，从而实现类似 GPS 定位系统一样的功能。

step 01 打开记事本文件，在其中输入如下代码：

```
<!DOCTYPE html>
<html>
<head>
<title>持续获取当前的位置</title>
</head>
<body>
<p id="demo">获取当前位置：</p>
<button onclick="getLocation()">定位当前位置</button>
<script>
var x=document.getElementById("demo");
function getLocation()
{
```

```
    if (navigator.geolocation)
    {
        navigator.geolocation.watchPosition(showPosition);
    }
    else
    {
        x.innerHTML="该浏览器不支持获取地理位置。";
    }
}
function showPosition(position)
{
    x.innerHTML="纬度: " + position.coords.latitude +
    "<br>经度: " + position.coords.longitude;
}
</script>
</body>
</html>
```

step 02 使用 IE 11.0 打开网页文件，如图 13-5 所示。单击“定位当前位置”按钮，即可获取目前的位置，如图 13-6 所示。用户移动位置后，再次单击“定位当前位置”按钮，就会发现坐标发生了变化。



图 13-5 程序运行结果



图 13-6 获取当前位置

13.5 高 手 解 惑

疑问 1：使用 HTML 5 Geolocation API 获得的用户地理位置一定精准无误吗？

答：不一定精准，因为该特性可能侵犯用户的隐私，除非用户同意，否则用户位置信息是不可用的。

疑问 2：地理位置 API 可以在国际空间站上使用吗？可以在月球上或者其他星球上用吗？

答：地理位置标准是这样阐述的：“地理坐标参考系的属性值来自大地测量系统(World Geodetic System (2d) [WGS84])。不支持其他参考系。”国际空间站位于地球轨道上，所以宇航员可以使用经纬度和海拔来描述其位置。但是，大地测量系统是以地球为中心的，因此也就不能使用这个系统来描述月球或者其他星球的位置了。



第 14 章 Web 存储和通信技术

Web Storage 是 HTML 5 引入的一个非常重要的功能，可以在客户端本地存储数据，类似 HTML 4 的 Cookie，但可实现的功能要比 Cookie 强大得多。Cookie 大小被限制在 4KB，而 Web Storage 官方建议为每个网站 5MB。另外，Web 通信技术可以更好地完成跨域数据的通信，以及 Web 即时通信应用的实现，如 Web QQ 等。



14.1 认识 Web 存储

在 HTML 5 标准之前，Web 存储信息需要 Cookie 来完成，但是 Cookie 不适合大量数据的存储，因为它们由每个对服务器的请求来传递，这使得 Cookie 速度很慢而且效率也不高。为此，在 HTML 5 中，Web 存储 API 为用户如何在计算机或设备上存储用户信息做了数据标准的定义。

14.1.1 本地存储和 Cookies 的区别

本地存储和 Cookies 扮演着类似的角色，但是它们有根本的区别。

(1) 本地存储是仅存储在用户的硬盘上，并等待用户读取，而 Cookies 是在服务器上读取。

(2) 本地存储仅供客户端使用，如果需要服务器端根据存储数值做出反应，就应该使用 Cookies。

(3) 读取本地存储不会影响到网络带宽，但是使用 Cookies 将会发送到服务器，这样会影响到网络带宽，无形中增加了成本。

(4) 从存储容量上看，本地存储可存储多达 5MB 的数据，而 Cookies 最多只能存储 4KB 的数据信息。

14.1.2 Web 存储方法

在 HTML 5 标准中，提供了以下两种在客户端存储数据的新方法。

(1) sessionStorage。sessionStorage 是基于 session 的数据存储，在关闭或者离开网站后，数据将会被删除，也被称为会话存储。

(2) localStorage。localStorage 是没有时间限制的数据存储，也被称为本地存储。

与会话存储不同，本地存储将在用户计算机上永久保持数据信息。关闭浏览器窗口后，如果再次打开该站点，将可以检索所有存储在本地的数据。

在 HTML 5 中，数据不是由每个服务器请求传递的，而是只有在请求时使用数据，这样的话，存储大量数据时不会影响网站性能。对于不同的网站，数据存储于不同的区域，并且一个网站只能访问其自身的数据。



HTML 5 使用 JavaScript 来存储和访问数据。为此，建议用户可以多了解一下 JavaScript 的基本知识。

14.2 使用 HTML 5 Web Storage API

使用 HTML 5 Web Storage API 技术，可以实现很好的本地存储。

14.2.1 测试浏览器的支持情况

Web Storage 在各大主流浏览器中都得到了支持，但是为了兼容老的浏览器，还是要检查一下是否可以使用这项技术，主要有以下两种方法。

1. 通过检查 Storage 对象是否存在

第一种方式：通过检查 Storage 对象是否存在，来检查浏览器是否支持 Web Storage。代码如下：

```
if(typeof(Storage)!=="undefined")
{
    // 是的! 支持 localStorage  sessionStorage 对象!
    // 一些代码.....
} else {
    // 抱歉! 不支持 web 存储。
}

if(typeof(Storage)!=="undefined"){
//是的! 支持 localStorage  sessionStorage 对象!
//一些代码.....
} else {
//抱歉! 不支持 web 存储。
}
```

2. 分别检查各自的对象

第二种方式：分别检查各自的对象。例如：检查 localStorage 是否支持。代码如下：

```
if (typeof(localStorage) == 'undefined' ) {
alert('Your browser does not support HTML 5 localStorage. Try upgrading.');
} else {
//是的! 支持 localStorage  sessionStorage 对象!
//一些代码.....
}
或者:
if('localStorage' in window && window['localStorage'] !== null){
//是的! 支持 localStorage  sessionStorage 对象!
//一些代码.....
} else {
alert('Your browser does not support HTML 5 localStorage. Try upgrading.');
}
或者
if (!!localStorage) {
//是的! 支持 localStorage  sessionStorage 对象!
//一些代码....
} else {
alert('您的浏览器不支持localStorage  sessionStorage 对象!');
}
```

14.2.2　案例 1——使用 sessionStorage 方法创建对象

sessionStorage 方法针对一个 session 进行数据存储。在用户关闭浏览器窗口后，数据会被自动删除。

创建一个 sessionStorage 方法的基本语法格式如下：

```
<script type="text/javascript">
sessionStorage.abc="  ";
</script>
```

1. 创建对象

【例 14.1】使用 sessionStorage 方法创建对象(案例文件：ch14\14.1.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
sessionStorage.name="努力过好每一天！";
document.write(sessionStorage.name);
</script>
</body>
</html>
```

在 Firefox 53.0 中浏览，效果如图 14-1 所示。即可看到使用 sessionStorage 方法创建的对象内容显示在网页中。



图 14-1　使用 sessionStorage 方法创建对象

2. 制作网站访问记录计数器

下面继续使用 sessionStorage 方法来做一个实例，主要制作记录用户访问网站次数的计数器。

【例 14.2】制作网站访问记录计数器(案例文件：ch14\14.2.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
if (sessionStorage. count)
 {
 sessionStorage.count=Number(sessionStorage.count) +1;
```

```
 }
else
 {
 sessionStorage. count=1;
 }
document.write("您访问该网站的次数为: " + sessionStorage.count);
</script>
</body>
</html>
```

在 Firefox 53.0 中浏览，效果如图 14-2 所示。如果用户刷新一次页面，计数器的数值将进行加 1。



图 14-2 使用 sessionStorage 方法创建计数器



如果用户关闭浏览器窗口，再次打开该网页，计数器将重置为 1。

14.2.3 案例 2——使用 localStorage 方法创建对象

与 sessionStorage 方法不同，localStorage 方法存储的数据没有时间限制。也就是说，网页浏览者关闭网页很长一段时间后，再次打开此网页时，数据依然可用。

创建一个 localStorage 方法的基本语法格式如下：

```
<script type="text/javascript">
localStorage.abc="  ";
</script>
```

1. 创建对象

【例 14.3】使用 localStorage 方法创建对象(案例文件：ch14\14.3.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
localStorage.name="学习 HTML 5 最新的技术：Web 存储";
document.write(localStorage.name);
</script>
</body>
</html>
```

在 Firefox 53.0 中浏览，效果如图 14-3 所示。可以看到，使用 localStorage 方法创建的对象内容显示在网页中。



图 14-3　使用 localStorage 方法创建对象

2. 制作网站访问记录计数器

下面仍然使用 localStorage 方法来制作记录用户访问网站次数的计数器。用户可以清楚地看到 localStorage 方法和 sessionStorage 方法的区别。

【例 14.4】制作网站访问记录计数器(案例文件：ch14\14.4.html)。

```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
if (localStorage.count)
 {
 localStorage.count=Number(localStorage.count) +1;
 }
else
 {
 localStorage.count=1;
 }
document.write("您访问该网站的次数为: " + localStorage.count");
</script>
</body>
</html>
```

在 Firefox 53.0 中浏览，效果如图 14-4 所示。如果用户刷新一次页面，计数器的数值将进行加 1；如果用户关闭浏览器窗口，再次打开该网页，计数器会继续上一次计数，而不会重置为 1。



图 14-4　使用 localStorage 方法创建计数器

14.2.4　案例 3——Web Storage API 的其他操作

Web Storage API 的 localStorage 和 sessionStorage 对象除了以上基本应用外，还有以下两

个方面的应用。

1. 清空 localStorage 数据

localStorage 的 clear()函数用于清空同源的本地存储数据，比如 localStorage.clear()，它将删除所有本地存储的 localStorage 数据。

而 Web Storage 的另外一部分 Session Storage 中的 clear()函数只清空当前会话存储的数据。

2. 遍历 localStorage 数据

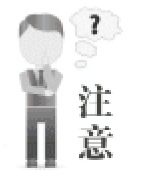
遍历 localStorage 数据可以查看 localStorage 对象保存的全部数据信息。在遍历过程中，需要访问 localStorage 对象的另外两个属性 length 与 key。length 表示 localStorage 对象中保存数据的总量，key 表示保存数据时的键名项，该属性常与索引号(index)配合使用，表示第几条键名对应的数据记录。其中，索引号(index)以 0 值开始，如果取第 3 条键名对应的数据，index 值应该为 2。

取出数据并显示数据内容的代码如下：

```
function showInfo(){
   var array=new Array();
   for(var i=0;i
   //调用 key 方法获取 localStorage 中数据对应的键名
   //如这里键名是从 test1 开始递增到 testN 的，那么 localStorage.key(0)对应 test1
   var getKey=localStorage.key(i);
   //通过键名获取值，这里的值包括内容和日期
   var getVal=localStorage.getItem(getKey);
   //array[0]就是内容，array[1]是日期
   array=getVal.split(",");
   }
}
```

获取并保存数据的代码如下：

```
var storage = window.localStorage; f
or (var i=0, len = storage.length; i  <  len; i++){
var key = storage.key(i);
var value = storage.getItem(key);
console.log(key + "=" + value); }
```



由于 localStorage 不仅仅是存储了这里所添加的信息，可能还存在其他信息。但是那些信息的键名也是以递增数字形式表示的。这样，如果这里也用纯数字就可能覆盖另外一部分的信息。所以建议键名都用独特的字符区分开，这里在每个 ID 前加上 test 以示区别。

14.2.5 案例 4——使用 JSON 对象存取数据

在 HTML 5 中可以使用 JSON 对象来存取一组相关的对象。使用 JSON 对象可以收集一组用户输入信息，然后创建一个 Object 来囊括这些信息，之后用一个 JSON 字符串来表示这个 Object，然后把 JSON 字符串存放在 localStorage 中。当用户检索指定名称时，会自动用该

名称去 localStorage 取得对应的 JSON 字符串，将字符串解析到 Object 对象，然后依次提取对应的信息，并构造 HTML 文本输入显示。

【例 14.5】使用 JSON 对象存取数据(案例文件：ch14\14.5.html)。

下面就来列举一个简单的案例，来介绍如何使用 JSON 对象存取数据。具体操作步骤如下。

step 01 新建一个记事本文件，具体代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>使用 JSON 对象存取数据</title>
<script type="text/javascript" src="objectStorage.js"></script>
</head>
<body>
<h3>使用 JSON 对象存取数据</h3>
<h4>填写待存取信息到表格中</h4>
<table>
<tr><td>用户名:</td><td><input type="text" id="name"></td></tr>
<tr><td>E-mail:</td><td><input type="text" id="email"></td></tr>
<tr><td>联系电话:</td><td><input type="text" id="phone"></td></tr>
<tr><td></td><td><input type="button" value="保存" onclick="saveStorage();">
</td></tr>
</table>
<hr>
<h4> 检索已经存入 localStorage 的 json 对象，并且展示原始信息</h4>
<p>
<input type="text" id="find">
<input type="button" value="检索" onclick="findStorage('msg');">
</p>
<!-- 下面这块用于显示被检索到的信息文本 -->
<p id ="msg"></p>
</body>
</html>
```

step 02 使用 Firefox 53.0 浏览保存的 html 文件，效果如图 14-5 所示。

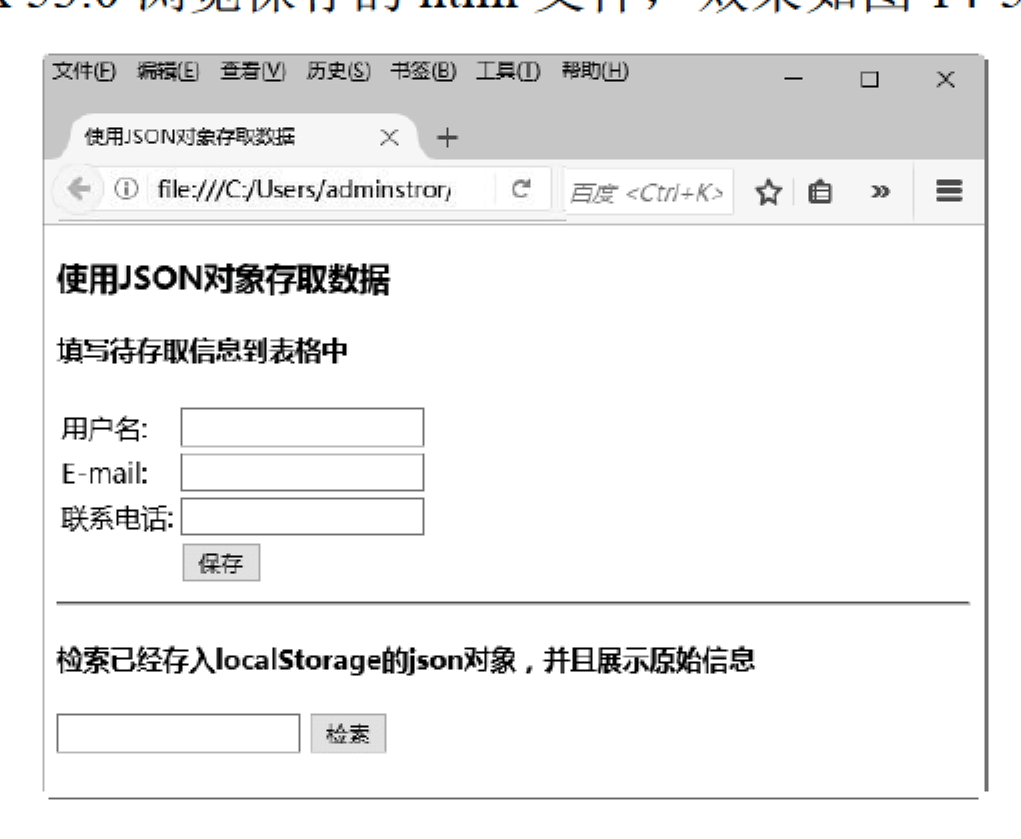


图 14-5　创建存取对象表格

step 03 案例中用到了 JavaScript 脚本，其中包含两个函数，一个是存数据，另一个是取数据。具体的 JavaScript 脚本代码如下：

```
function saveStorage(){            //创建一个 js 对象，用于存放当前从表单获得的数据
```

```
var data = new Object;          //将对象的属性值名依次和用户输入的属性值关联起来
data.user=document.getElementById("user").value;
data.mail=document.getElementById("mail").value;
data.tel=document.getElementById("tel").value;
//创建一个 json 对象，让其对应 html 文件中创建的对象的字符串数据形式
var str = JSON.stringify(data);
//将 json 对象存放到 localStorage 上，key 为用户输入的 NAME，value 为这个 json 字符串
localStorage.setItem(data.user,str);
console.log("数据已经保存！ 被保存的用户名为: "+data.user);
}
//从 localStorage 中检索用户输入的名称对应的 json 字符串，然后把 json 字符串解析为一组信
//息，并且打印到指定位置
function findStorage(id){        //获得用户的输入，是用户希望检索的名字
var requiredPersonName = document.getElementById("find").value;
//以这个检索的名字来查找 localStorage，得到了 json 字符串
var str=localStorage.getItem(requiredPersonName);
//解析这个 json 字符串得到 Object 对象
var data= JSON.parse(str);
//从 Object 对象中分离出相关属性值，然后构造要输出的 HTML 内容
var result="用户名:"+data.user+'<br>';
result+="E-mail:"+data.mail+'<br>';
result+="联系电话:"+data.tel+'<br>';          //取得页面上要输出的容器
var target = document.getElementById(id);   //用刚才创建的 HTML 内容来填充这个容器
target.innerHTML = result;
}
```

step 04 将 JS 文件和 HTML 文件放在同一目录下，再次打开网页，在表单中依次输入相关内容，单击“保存”按钮，如图 14-6 所示。

step 05 在“检索”文本框中输入已经保存的信息的用户名，单击“检索”按钮，则在页面下方自动显示保存的用户信息，如图 14-7 所示。

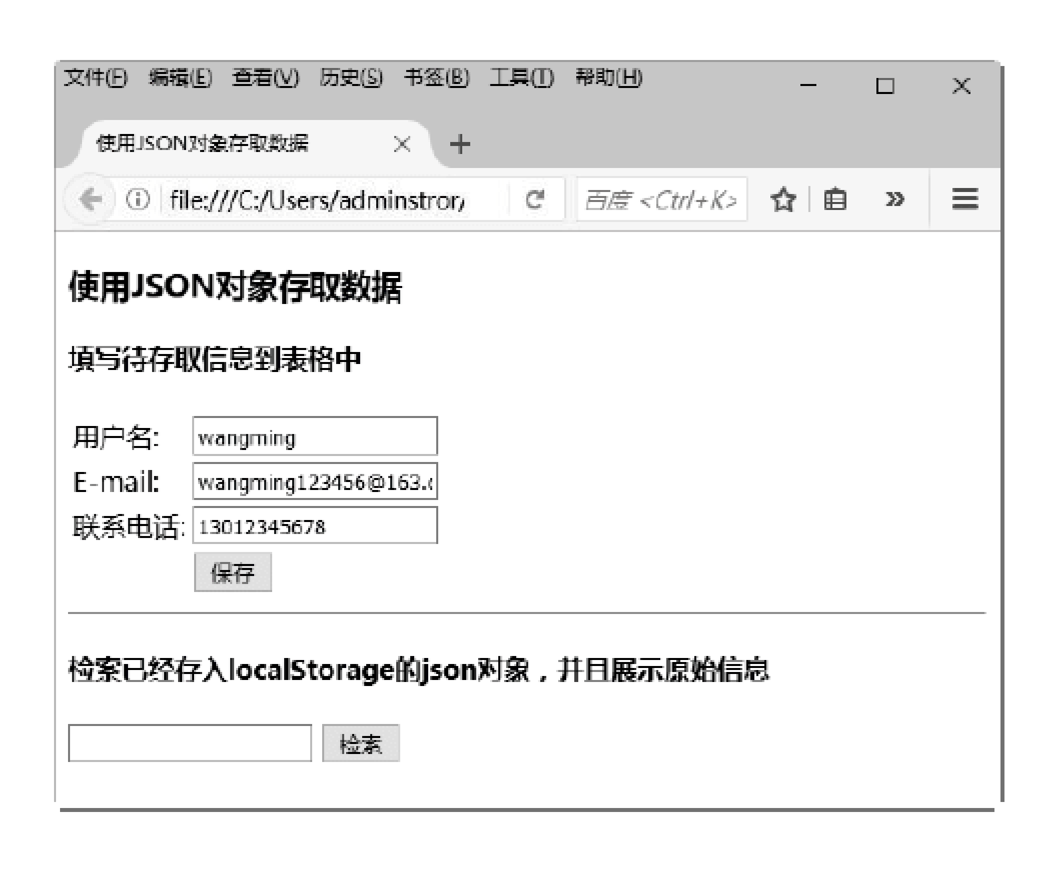


图 14-6　输入表格内容

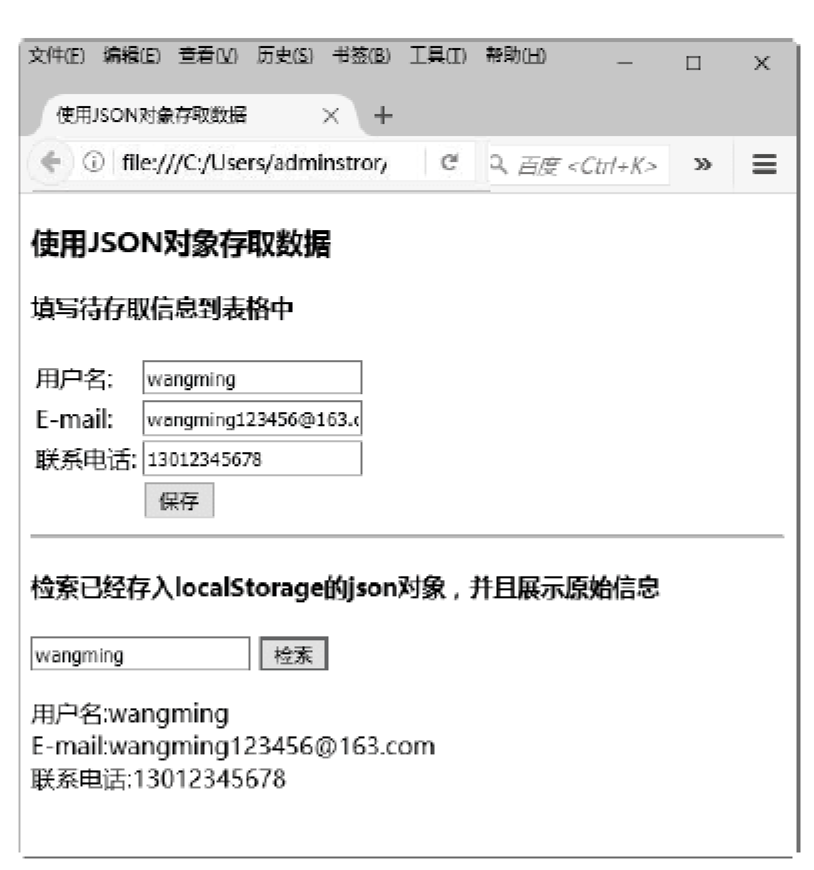


图 14-7　检索数据信息

14.3　目前浏览器对 Web 存储的支持情况

不同的浏览器版本对 Web 存储技术的支持情况是不同的。如表 14-1 所示是常见浏览器对

Web 存储的支持情况。

表 14-1　常见浏览器对 Web 存储的支持情况

浏览器名称	支持 Web 存储技术的版本
Internet Explorer	Internet Explorer 8 及更高版本
Firefox	Firefox 3.6 及更高版本
Opera	Opera 10.0 及更高版本
Safari	Safari 4 及更高版本
Chrome	Chrome 5 及更高版本
Android	Android 2.1 及更高版本

14.4　跨文档消息传输

利用跨文档消息传输功能，可以在不同域、端口或网页文档之间进行消息的传递。

14.4.1　跨文档消息传输的基本知识

利用跨文档消息传输可以实现跨域的数据推动，使服务器端不再被动地等待客户端的请求。只要客户端与服务器端建立了一次链接之后，服务器端就可以在需要的时候，主动地将数据推送到客户端，直到客户端显示关闭这个链接。

HTML 5 提供了在网页文档之间互相接收与发送消息的功能。使用这个功能，只要获取到网页所在页面对象的实例，不仅同域的 Web 网页之间可以互相通信，甚至可以实现跨域通信。

想要接收从其他文档那里发过来的消息，就必须对文档对象的 message 时间进行监视，实现代码如下。

```
window.addEventListener("message", function(){…}, false)
```

想要发送消息，可以使用 window 对象的 postMessage 方法来实现，该方法的实现代码如下：

```
otherWindow.postMessage(message, targetOrigin)
```



postMessage 是 HTML 5 为解决跨文档通信，特别引入的一个新的 API，目前支持这个 API 的浏览器有：IE(8.0 以上)、Firefox、Opera、Safari 和 Chrome。

postMessage 允许页面中的多个 iframe/window 的通信，postMessage 也可以实现 ajax 直接跨域，不通过服务器端代理。

14.4.2　案例 5——跨文档通信应用测试

下面介绍一个跨文档通信的应用案例，其中主要使用 postMessage 的方法来实现该案例。

具体操作步骤如下。

需要创建两个文档来实现跨文档的访问，名称分别为 14.6.html 和 14.7.html。

step 01 打开记事本文件，在其中输入以下代码，以创建用于实现信息发送的 14.6.html 文档。具体代码如下：

```
<!DOCTYPE HTML>
<html>
<head>
  <title>跨域文档通信 1</title>
</head>
<script type="text/javascript">
  window.onload = function() {
    document.getElementById('title').innerHTML = '页面在' + document.location.host
    + '域中，且每过 1 秒向 14.7.html 文档发送一个消息！';
    //定时向另外一个不确定域的文件发送消息
    setInterval(function(){
       var message = '消息发送测试！   ' + (new Date().getTime());
       window.parent.frames[0].postMessage(message, '*');
    },1000);
  };
</script>
<body>
<div id="title"></div>
</body>
</html>
```

step 02 保存记事本文件，然后使用浏览器打开该文件，最终的效果如图 14-8 所示。



图 14-8 程序运行结果

step 03 打开记事本文件，在其中输入以下代码，以创建用于实现信息监听的 14.7.html 文档。具体代码如下：

```
<!DOCTYPE HTML>
<html>
<head>
  <title>跨域文档通信 2</title>
</head>

<script type="text/javascript">
  window.onload = function() {
```

```
    document.getElementById('title').innerHTML = '页面在' +
document.location.host + '域中，且每过 1 秒向 14.6.html 文档发送一个消息！';
    //定时向另外一个不同域的 iframe 发送消息
    setInterval(function(){
        var message = '消息发送测试！   ' + (new Date().getTime());
        window.parent.frames[0].postMessage(message, '*');
    },1000);

    var onmessage = function(e) {
       var data = e.data,p = document.createElement('p');
       p.innerHTML = data;
       document.getElementById('display').appendChild(p);
    };
    //监听 postMessage 消息事件
    if (typeof window.addEventListener != 'undefined') {
      window.addEventListener('message', onmessage, false);
    } else if (typeof window.attachEvent != 'undefined') {
      window.attachEvent('onmessage', onmessage);
    }

  };

</script>

<body>
<div id="title"></div>
<br>
<div id="display"></div>
</body>
</html>
```

step 04 在 Firefox 53.0 中运行 14.8.html 文件，效果如图 14-9 所示。



图 14-9　程序运行结果

在 14.6.html 文件中的“window.parent.frames[0].postMessage(message, '*');”语句中的“*”表示不对访问的域进行判断。如果要加入特定域的限制，可以将代码改为“window.parent.frames[0].postMessage(message, 'url');”其中的 url 必须为完整的网站域名格式。而在信息监听接收方的 onmessage 中需要追加一个判断语句“if(event.origin !== 'url') return;”。



由于在实际通信时，应当实现双向通信，所以，在编写代码时，每一个文档中都应该具有发送信息和监听接收信息的模块。

14.5 WebSocket API

HTML 5 中有一个很实用的新特性：WebSocket。使用 WebSocket 可以在没有 Ajax 请求的情况下与服务器端对话。

14.5.1 什么是 WebSocket API

WebSocket API 是下一代客户端-服务器的异步通信方法。该通信取代了单个的 TCP 套接字，使用 WS 或 WSS 协议，可用于任意的客户端和服务器程序。WebSocket 目前由 W3C 进行标准化。WebSocket 已经得到 Firefox 4、Chrome 4、Opera 10.70 及 Safari 5 等浏览器的支持。

WebSocket API 最伟大之处在于服务器和客户端可以在给定的时间范围内的任意时刻，相互推送信息。WebSocket 并不限于以 Ajax(或 XHR)方式通信，因为 Ajax 技术需要客户端发起请求，而 WebSocket 服务器和客户端可以彼此相互推送信息；XHR 受到域的限制，而 WebSocket 允许跨域通信。

Ajax 技术很聪明的一点是没有涉及要使用的方式。WebSocket 为指定目标创建，用于双向推送消息。

14.5.2 WebSocket 通信基础

1. 产生 WebSockets 的背景

随着即时通信系统的普及，基于 Web 的实时通信也变得普及，如新浪微博的评论、私信的通知，腾讯的 Web QQ 等。

在 WebSocket 出现之前，一般通过两种方式来实现 Web 实时应用：轮询机制和流技术，而其中的轮询机制又可分为普通轮询和长轮询(Coment)，分别介绍如下。

(1) 轮询。这是最早的一种实现实时 Web 应用的方案。客户端以一定的时间间隔向服务端发出请求，以频繁请求的方式来保持客户端和服务器端的同步。这种同步方案的缺点是，当客户端以固定频率向服务器发起请求的时候，服务器端的数据可能并没有更新，这样会带来很多无谓的网络传输，所以这是一种非常低效的实时方案。

(2) 长轮询。这是对定时轮询的改进和提高，目的是降低无效的网络传输。当服务器端没有数据更新的时候，连接会保持一段时间，直到数据或状态改变或者时间过期，通过这种机制来减少无效的客户端和服务器间的交互。当然，如果服务器端的数据变更非常频繁的话，这种机制和定时轮询比较起来没有本质上的性能的提高。

(3) 流。就是在客户端的页面使用一个隐藏的窗口向服务器端发出一个长连接的请求。服务器端接到这个请求后做出回应并不断更新连接状态以保证客户端和服务器端的连接不过期。通过这种机制可以将服务器端的信息源源不断地推向客户端。这种机制在用户体验上有

一点问题，需要针对不同的浏览器设计不同的方案来改进用户体验，同时这种机制在并发比较大的情况下，对服务器端的资源是一个极大的考验。

但是上述 3 种方式实际看来都不是真正的实时通信技术，只是相对地模拟出了实时的效果。这种效果的实现对编程人员来说无疑增加了复杂性，对于客户端和服务器端的实现都需要复杂的 HTTP 链接设计来模拟双向的实时通信。这种复杂的实现方法制约了应用系统的扩展性。

基于上述弊端，在 HTML 5 中增加了实现 Web 实时应用的技术：Web Socket。Web Socket 通过浏览器提供的 API 真正实现了具备像 C/S 架构下的桌面系统的实时通信能力。其原理是使用 JavaScript 调用浏览器的 API 发出一个 WebSocket 请求至服务器，经过一次握手，和服务器建立了 TCP 通信，因为它本质上是一个 TCP 连接，所以数据传输的稳定性强和数据传输量比较小。由于 HTML 5 中 WebSockets 的实用，使其具备了 Web TCP 的称号。

2. WebSocket 技术的实现方法

WebSocket 技术本质上是一个基于 TCP 的协议技术。其建立通信链接的具体操作步骤如下。

step 01 为了建立一个 WebSocket 连接，客户端的浏览器首先要向服务器发起一个 HTTP 请求，这个请求和通常的 HTTP 请求有所差异，除了包含一般的头信息外，还有一个附加的信息 Upgrade: WebSocket，表明这是一个申请协议升级的 HTTP 请求。

step 02 服务器端解析这些附加的头信息，经过验证后，产生应答信息返回给客户端。

step 03 客户端接收返回的应答信息，建立与服务器端的 WebSocket 连接，之后双方就可以通过这个连接通道自由地传递信息，并且这个连接会持续存在，直到客户端或者服务器端的某一方主动地关闭连接。

WebSocket 技术，目前还是属于比较新的技术，其版本更新较快。目前的最新版本基本上可以被 Chrome、FireFox、Opera 和 IE(9.0 以上)等浏览器支持。

在建立实时通信时，客户端发到服务器的内容如下：

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat8.Sec-WebSocket-Version: 13
```

从服务器返回到客户端的内容如下：

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```



其中的 Upgrade:WebSocket 表示这是一个特殊的 HTTP 请求，请求的目的就是要将客户端和服务器端的通信协议从 HTTP 协议升级到 WebSocket 协议。其中客户端的 Sec-WebSocket-Key 和服务器端的 Sec-WebSocket-Accept 就是重要的握手认证信息，实现握手后才可以进一步地进行信息的发送和接收。

14.5.3 案例 6——服务器端使用 Web Socket API

在实现 WebSocket 实时通信时，需要使客户端和服务器端建立链接，需要配置相应的内容。一般构建链接握手时，客户端的内容浏览器都可以代劳完成，主要实现的是服务器端的内容。下面来看一下 WebSockets API 的具体使用方法。

服务器端需要编程人员自己来实现。目前市场上可以直接使用的开源方法比较多，主要有以下 5 种。

(1) Kaazing WebSocket Gateway：是一个 Java 实现的 WebSocket Server。

(2) mod_pywebsocket：是一个 Python 实现的 WebSocket Server。

(3) Netty：是一个 Java 实现的网络框架，其中包括了对 WebSocket 的支持。

(4) node.js：是一个 Server 端的 JavaScript 框架，提供了对 WebSocket 的支持。

(5) WebSocket4Net：是一个.net 的服务器端实现。

除了使用以上开源的方法外，自己编写一个简单的服务器端也是可以的。其中服务器端需要实现握手、接收和发送 3 个内容。

下面就来详细介绍一下操作方法。

1. 握手

在实现握手时需要通过 Sec-WebSocket 信息来实现验证。使用 Sec-WebSocket-Key 和一个随机值构成一个新的 key 串，然后将新的 key 串进行 SHA1 编码，生成一个由多组两位 16 进制数构成的加密串；最后再把加密串进行 base64 编码生成最终的 key，这个 key 就是 Sec-WebSocket- Accept。

实现 Sec-WebSocket-Key 运算的实例代码如下：

```
/// <summary>
/// 生成 Sec-WebSocket-Accept
/// </summary>
/// <param name="handShakeText">客户端握手信息</param>
/// <returns>Sec-WebSocket-Accept</returns>
private static string GetSecKeyAccetp(byte[] handShakeBytes,int bytesLength)
{
    string handShakeText = Encoding.UTF8.GetString(handShakeBytes, 0, bytesLength);
    string key = string.Empty;
    Regex r = new Regex(@"Sec\-WebSocket\-Key:(.*?)\r\n");
    Match m = r.Match(handShakeText);
    if (m.Groups.Count != 0)
    {
    key = Regex.Replace(m.Value, @"Sec\-WebSocket\-Key:(.*?)\r\n", "$1").Trim();
    }
```

```
    byte[] encryptionString = SHA1.Create().ComputeHash(Encoding.ASCII.
GetBytes(key + "258EAFA5-E914-47DA-95CA-C5AB0DC85B11"));
    return Convert.ToBase64String(encryptionString);
}
```

2. 接收

如果握手成功，将会触发客户端的 onOpen 事件，进而解析接收的客户端信息。在进行数据信息解析时，会将数据以字节和比特的方式拆分，并按照以下规则进行解析。

(1) 第 1byte。

1bit：frame-fin，x0 表示该 message 后续还有 frame；x1 表示是 message 的最后一个 frame。

3bit：分别是 frame-rsv1、frame-rsv2 和 frame-rsv3，通常都是 x0。

4bit：frame-opcode，x0 表示是延续 frame；x1 表示文本 frame；x2 表示二进制 frame；x3-7 保留给非控制 frame；x8 表示关闭连接；x9 表示 ping；xA 表示 pong；xB-F 保留给控制 frame。

(2) 第 2byte。

1bit：Mask，1 表示该 frame 包含掩码；0 表示无掩码。

7bit、7bit+2byte、7bit+8byte：7bit 取整数值，若为 0～145，则是负载数据长度；若是 146 表示后两个 byte 取无符号 16 位整数值，是负载长度；147 表示后 8 个 byte，取 64 位无符号整数值，是负载长度。

(3) 第 3～6byte。这里假定负载长度为 0～145，并且 Mask 为 1，则这 4 个 byte 是掩码。

(4) 第 7-end byte。长度是上面取出的负载长度，包括扩展数据和应用数据两个部分，通常没有扩展数据；若 Mask 为 1，则此数据需要解码，解码规则为 1～4byte 掩码循环和数据 byte 做异或操作。

实现数据解析的代码如下：

```
/// <summary>
/// 解析客户端数据包
/// </summary>
/// <param name="recBytes">服务器接收的数据包</param>
/// <param name="recByteLength">有效数据长度</param>
/// <returns></returns>
private static string AnalyticData(byte[] recBytes, int recByteLength)
{
    if (recByteLength < 2) { return string.Empty; }
    bool fin = (recBytes[0] & 0x80) == 0x80; // 1bit，1 表示最后一帧
    if (!fin){
    return string.Empty;// 超过一帧暂不处理
    }
    bool mask flag = (recBytes[1] & 0x80) == 0x80; // 是否包含掩码
    if (!mask flag){
    return string.Empty;// 不包含掩码的暂不处理
    }
    int payload len = recBytes[1] & 0x7F; // 数据长度
    byte[] masks = new byte[4];
    byte[] payload_data;
```

```
    if (payload len == 146){
    Array.Copy(recBytes, 4, masks, 0, 4);
    payload len = (UInt16)(recBytes[2] << 8 | recBytes[3]);
    payload_data = new byte[payload_len];
    Array.Copy(recBytes, 8, payload data, 0, payload len);
    }else if (payload_len == 147){
    Array.Copy(recBytes, 10, masks, 0, 4);
    byte[] uInt64Bytes = new byte[8];
    for (int i = 0; i < 8; i++){
        uInt64Bytes[i] = recBytes[9 - i];
    }
    UInt64 len = BitConverter.ToUInt64(uInt64Bytes, 0);
    payload data = new byte[len];
    for (UInt64 i = 0; i < len; i++){
        payload data[i] = recBytes[i + 14];
    }
        }else{
    Array.Copy(recBytes, 2, masks, 0, 4);
    payload data = new byte[payload len];
    Array.Copy(recBytes, 6, payload data, 0, payload len);
        }
        for (var i = 0; i < payload len; i++){
    payload data[i] = (byte)(payload data[i] ^ masks[i % 4]);
        }
        return Encoding.UTF8.GetString(payload_data);56.}
```

3. 发送

服务器端接收并解析了客户端发来的信息后，要返回回应信息。服务器端发送的数据以 0x81 开头，紧接着是发送内容的长度，最后是内容的 byte 数组。

实现数据发送的代码如下：

```
/// <summary>
/// 打包服务器数据
/// </summary>
/// <param name="message">数据</param>
/// <returns>数据包</returns>
private static byte[] PackData(string message)
{
    byte[] contentBytes = null;
    byte[] temp = Encoding.UTF8.GetBytes(message);
    if (temp.Length < 146){
    contentBytes = new byte[temp.Length + 2];
    contentBytes[0] = 0x81;
    contentBytes[1] = (byte)temp.Length;
    Array.Copy(temp, 0, contentBytes, 2, temp.Length);
    }else if (temp.Length < 0xFFFF){
    contentBytes = new byte[temp.Length + 4];
    contentBytes[0] = 0x81;
    contentBytes[1] = 146;
    contentBytes[2] = (byte)(temp.Length & 0xFF);
    contentBytes[3] = (byte)(temp.Length >> 8 & 0xFF);
    Array.Copy(temp, 0, contentBytes, 4, temp.Length);
    }else{
```

```
// 暂不处理超长内容
    }
    return contentBytes;
}
```

14.5.4　案例 7——客户端使用 WebSocket API

一般浏览器提供的 API 就可以直接用来实现客户端的握手操作了，在应用时直接使用 JavaScript 来调用即可。

客户端调用浏览器 API，实现握手操作的 JavaScript 代码如下：

```
var wsServer = 'ws://localhost:8888/Demo'; //服务器地址
var websocket = new WebSocket(wsServer);   //创建 WebSocket 对象
websocket.send("hello");                   //向服务器发送消息
alert(websocket.readyState);               //查看 websocket 当前状态
websocket.onopen = function (evt) {        //已经建立连接
};
websocket.onclose = function (evt) {       //已经关闭连接
};
websocket.onmessage = function (evt) {     //收到服务器消息，使用 evt.data 提取
};
websocket.onerror = function (evt) {       //产生异常
};
```

14.6　综合案例——制作简单 Web 留言本

使用 Web Storage 的功能可以用来制作 Web 留言本。具体制作方法如下。

step 01 构建页面框架，代码如下：

```
<!DOCTYPE html>
<html>
<head>
<title>本地存储技术之 Web 留言本</title>
</head>
<body onload="init()">
</body>
</html>
```

step 02 添加页面文件，主要由表单构成，包括单行文字表单和多行文本表单。代码如下：

```
    <h1>Web 留言本</h1>
    <table>
        <tr>
            <td>用户名</td>
            <td><input type="text" name="name" id="name" /></td>
        </tr>
        <tr>
            <td>留言</td>
            <td><textarea name="memo" id="memo" cols ="50" rows = "5">
```

```
</textarea></td>
        </tr>
        <tr>
            <td></td>
            <td>
                <input type="submit" value="提交" onclick="saveData()" />
            </td>
        </tr>
    </table>
    <ht>
    <table id="datatable" border="1"></table>
    <p id="msg"></p>
```

step 03 为了执行本地数据库的保存及调用功能，需要插入数据库的脚本代码，具体内容如下：

```
<script>
var datatable = null;
var db = openDatabase("MyData","1.0","My Database",2*1024*1024);
function init()
{
    datatable = document.getElementById("datatable");
    showAllData();
}
function removeAllData(){
    for(var i = datatable.childNodes.length-1;i>=0;i--){
        datatable.removeChild(datatable.childNodes[i]);
    }
    var tr = document.createElement('tr');
    var th1 = document.createElement('th');
    var th2 = document.createElement('th');
    var th3 = document.createElement('th');
    th1.innerHTML = "用户名";
    th2.innerHTML = "留言";
    th3.innerHTML = "时间";
    tr.appendChild(th1);
    tr.appendChild(th2);
    tr.appendChild(th3);
    datatable.appendChild(tr);
}
function showAllData()
{
    db.transaction(function(tx){
        tx.executeSql('create table if not exists MsgData(name TEXT,message
            TEXT,time INTEGER)',[]);
        tx.executeSql('select * from MsgData',[],function(tx,rs){
            removeAllData();
            for(var i=0;i<rs.rows.length;i++){
                showData(rs.rows.item(i));
            }
        });
    });
}
function showData(row){
```

```
    var tr=document.createElement('tr');
    var td1 = document.createElement('td');
    td1.innerHTML = row.name;
    var td2 = document.createElement('td');
    td2.innerHTML = row.message;
    var td3 = document.createElement('td');
    var t = new Date();
    t.setTime(row.time);
    ttd3.innerHTML = t.toLocaleDateString() + " " + t.toLocaleTimeString();
    tr.appendChild(td1);
    tr.appendChild(td2);
    tr.appendChild(td3);
    datatable.appendChild(tr);
}
function addData(name,message,time) {
    db.transaction(function(tx){
        tx.executeSql('insert into MsgData values(?,?,?)',[name,message,
            time],functionx,rs){
            alert("提交成功。");
        },function(tx,error){
            alert(error.source+"::"+error.message);
        });
    });
} // End of addData
function saveData() {
    var name = document.getElementById('name').value;
    var memo = document.getElementById('memo').value;
    var time = new Date().getTime();
    addData(name,memo,time);
    showAllData();
} // End of saveData
</script>
</head>
<body onload="init()">
    <h1>Web 留言本</h1>
    <table>
        <tr>
            <td>用户名</td>
            <td><input type="text" name="name" id="name" /></td>
        </tr>
        <tr>
            <td>留言</td>
            <td><textarea name="memo" id="memo" cols ="50" rows = "5">
</textarea></td>
        </tr>
        <tr>
            <td></td>
            <td>
                <input type="submit" value="提交" onclick="saveData()" />
            </td>
        </tr>
    </table>
    <ht>
    <table id="datatable" border="1"></table>
```

```
    <p id="msg"></p>
</body>
</html>
```

step 04 文件保存后，使用 Firefox 53.0 浏览页面，效果如图 14-10 所示。



图 14-10　Web 留言本

14.7　跟我学上机——编写简单的 WebSocket 服务器

前面学习了 WebSocket API 的原理及基本使用方法，提到在实现通信时关键要配置的是 WebSocket 服务器。下面介绍一个简单的 WebSocket 服务器编写方法。

为了实现操作，这里配合编写一个客户端文件，以测试服务器的实现效果。

step 01 首先编写客户端文件，其文件代码如下：

```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="UTF-8">
    <title>Web sockets test</title>
    <script src="jquery-min.js" type="text/javascript"></script>
    <script type="text/javascript">
        var ws;
        function ToggleConnectionClicked() {
          try {
          ws = new WebSocket("ws://192.168.1.101:1818/chat");//连接服务器
          ws.onopen = function(event){alert("已经与服务器建立了连接\r\n 当前连接状
              态: "+this.readyState);};
          ws.onmessage = function(event){alert("接收到服务器发送的数据:
              \r\n"+event.data);};
          ws.onclose = function(event){alert("已经与服务器断开连接\r\n 当前连接状
              态: "+this.readyState);};
          ws.onerror = function(event){alert("WebSocket 异常! ");};
              } catch (ex) {
          alert(ex.message);
          }
```

```
        };
        function SendData() {
        try{
        ws.send("jane");
        }catch(ex){
        alert(ex.message);
        }
        };
        function seestate(){
        alert(ws.readyState);
        }
    </script>
</head>
<body>
  <button id='ToggleConnection' type="button" onclick='ToggleConnectionClicked();
'>与服务器建立连接</button><br /><br />
    <button id='ToggleConnection' type="button" onclick='SendData();'>发送信
息：我的名字是 jane</button><br /><br />
    <button id='ToggleConnection' type="button" onclick='seestate();'>查看当
前状态</button><br /><br />
</body>
</html>
```

在 Opera 浏览器中预览，效果如图 14-11 所示。



图 14-11　程序运行结果



其中 ws.onopen、ws.onmessage、ws.onclose 和 ws.onerror 对应了 4 种状态的提示信息。在连接服务器时，需要在代码中指定服务器的链接地址，测试时将 IP 地址改为本机 IP 即可。

step 02 服务器程序可以使用.net 等实现编辑，编辑后服务器端的主程序代码如下：

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Security.Cryptography;
using System.Text;
using System.Text.RegularExpressions;
namespace WebSocket
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int port = 2828;
            byte[] buffer = new byte[1024];
            IPEndPoint localEP = new IPEndPoint(IPAddress.Any, port);
            Socket listener = new Socket(localEP.Address.AddressFamily,SocketType.
            Stream,ProtocolType.Tcp);
            try{
                listener.Bind(localEP);
                listener.Listen(10);
                Console.WriteLine("等待客户端连接....");
                Socket sc = listener.Accept();//接收一个连接
                Console.WriteLine("接收到了客户端: "+sc.RemoteEndPoint.ToString()+"
                    连接....");
                //握手
                int length = sc.Receive(buffer);//接收客户端握手信息
                sc.Send(PackHandShakeData(GetSecKeyAccetp(buffer,length)));
                Console.WriteLine("已经发送握手协议了....");
                //接收客户端数据
                Console.WriteLine("等待客户端数据....");
                length = sc.Receive(buffer);//接收客户端信息
                string clientMsg=AnalyticData(buffer, length);
                Console.WriteLine("接收到客户端数据: " + clientMsg);
                //发送数据
                string sendMsg = "您好, " + clientMsg;
                Console.WriteLine("发送数据: “"+sendMsg+"” 至客户端....");
                sc.Send(PackData(sendMsg));
                Console.WriteLine("演示 Over!");
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
        }
        …
        …
        …
        /// <summary>
        /// 打包服务器数据
        /// </summary>
        /// <param name="message">数据</param>
        /// <returns>数据包</returns>
        private static byte[] PackData(string message)
        {
            byte[] contentBytes = null;
            byte[] temp = Encoding.UTF8.GetBytes(message);
            if (temp.Length < 146){
                contentBytes = new byte[temp.Length + 2];
                contentBytes[0] = 0x81;
                contentBytes[1] = (byte)temp.Length;
               Array.Copy(temp, 0, contentBytes, 2, temp.Length);
            }else if (temp.Length < 0xFFFF){
                contentBytes = new byte[temp.Length + 4];
                contentBytes[0] = 0x81;
                contentBytes[1] = 146;
                contentBytes[2] = (byte)(temp.Length & 0xFF);
                contentBytes[3] = (byte)(temp.Length >> 8 & 0xFF);
```

```
                Array.Copy(temp, 0, contentBytes, 4, temp.Length);
            }else{
                // 暂不处理超长内容
            }
            return contentBytes;
        }
    }
}
```

内容较多，中间部分内容省略，编辑后保存服务器文件目录。

step 03 测试服务器和客户端的连接通信，首先打开服务器，运行本书配套资源中的“源代码\ch14\WebSocket 服务器\WebSocket-Server\WebSocket\obj\x86\Debug\WebSocket.exe”文件，提示等待客户端连接，效果如图 14-12 所示。

step 04 使用运行客户端文件(源代码\ch14\WebSocket 服务器\index.html)，效果如图 14-13 所示。



图 14-12　等待客户端连接

图 14-13　运行客户端文件

step 05 单击“与服务器建立连接”按钮，服务器端显示已经建立连接，客户端提示连接建立，且状态为 1，效果如图 14-14 所示。

step 06 单击“发送消息”按钮，自服务器端返回信息，提示“您好，jane”，如图 14-15 所示。

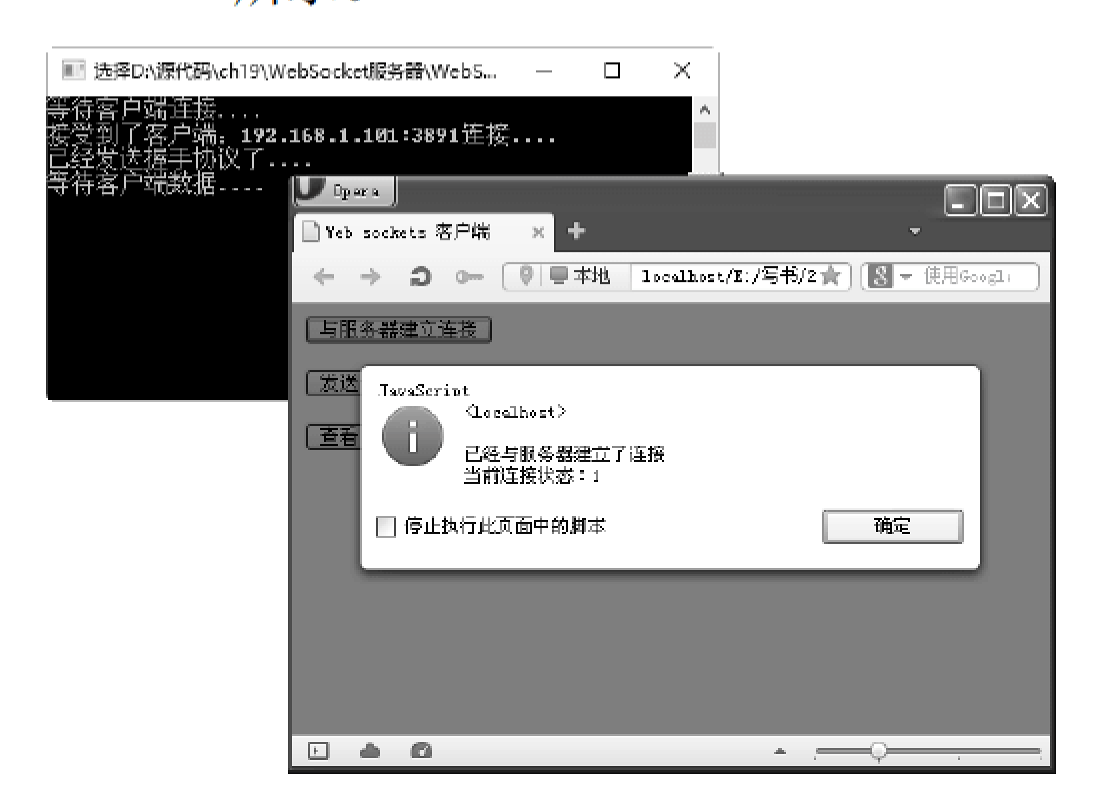


图 14-14　与服务器建立连接

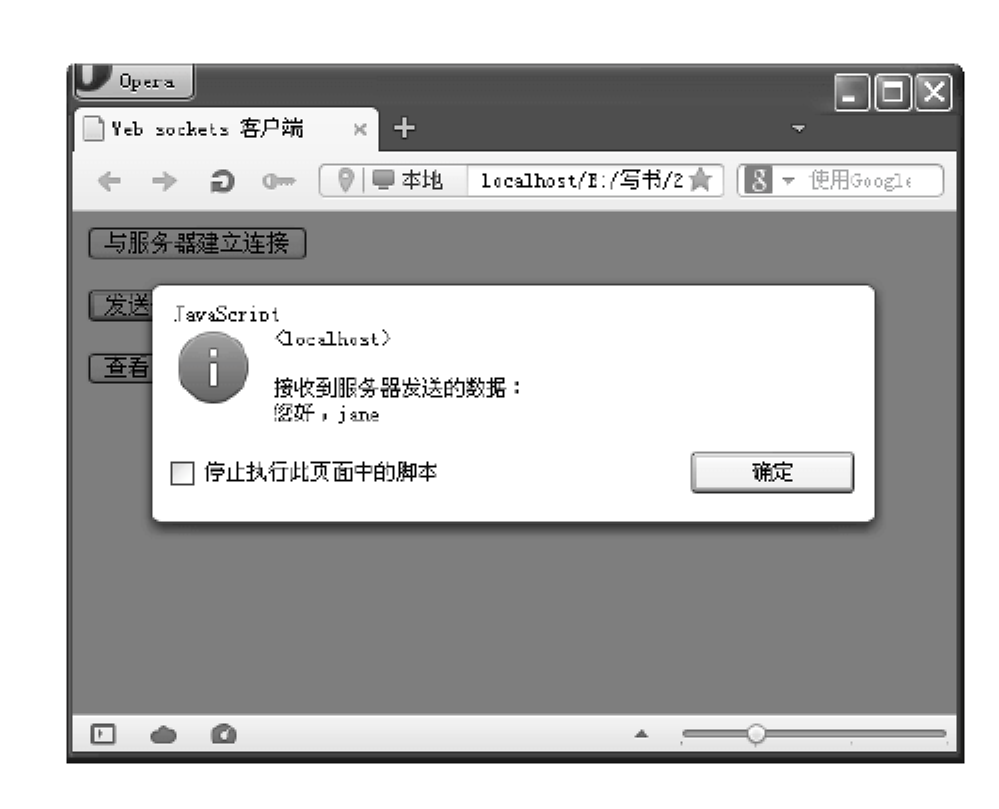


图 14-15　服务器端返回的信息

14.8 高手解惑

疑问 1：不同的浏览器可以读取同一个 Web 中存储的数据吗？

答：在 Web 存储时，不同的浏览器将存储在不同的 Web 存储库中。例如，如果用户使用的是 IE 浏览器，那么 Web 存储工作时，将所有数据将存储在 IE 的 Web 存储库中；如果用户再次使用火狐浏览器访问该站点，将不能读取 IE 浏览器存储的数据，可见每个浏览器的存储是分开并独立工作的。

疑问 2：离线存储站点时是否需要浏览者同意？

答：和地理定位类似，在网站使用 manifest 文件时，浏览器会提供一个权限提示，提示用户是否将离线设为可用，但不是每一个浏览器都支持这样的操作。

疑问 3：WebSocket 将会替代什么？

答：WebSockets 可以替代 Long Polling(PHP 服务端推送技术)。客户端发送一个请求到服务器，现在，服务器端并不会响应还没准备好的数据。它会保持连接的打开状态直到最新的数据准备就绪发送。之后客户端收到数据，然后发送另一个请求。好处在于减少任一连接的延迟，当一个连接已经打开时就不需要创建另一个新的连接。但是 Long-Polling 并不是什么花哨技术，它仍有可能发生请求暂停，因此会需要建立新的连接。

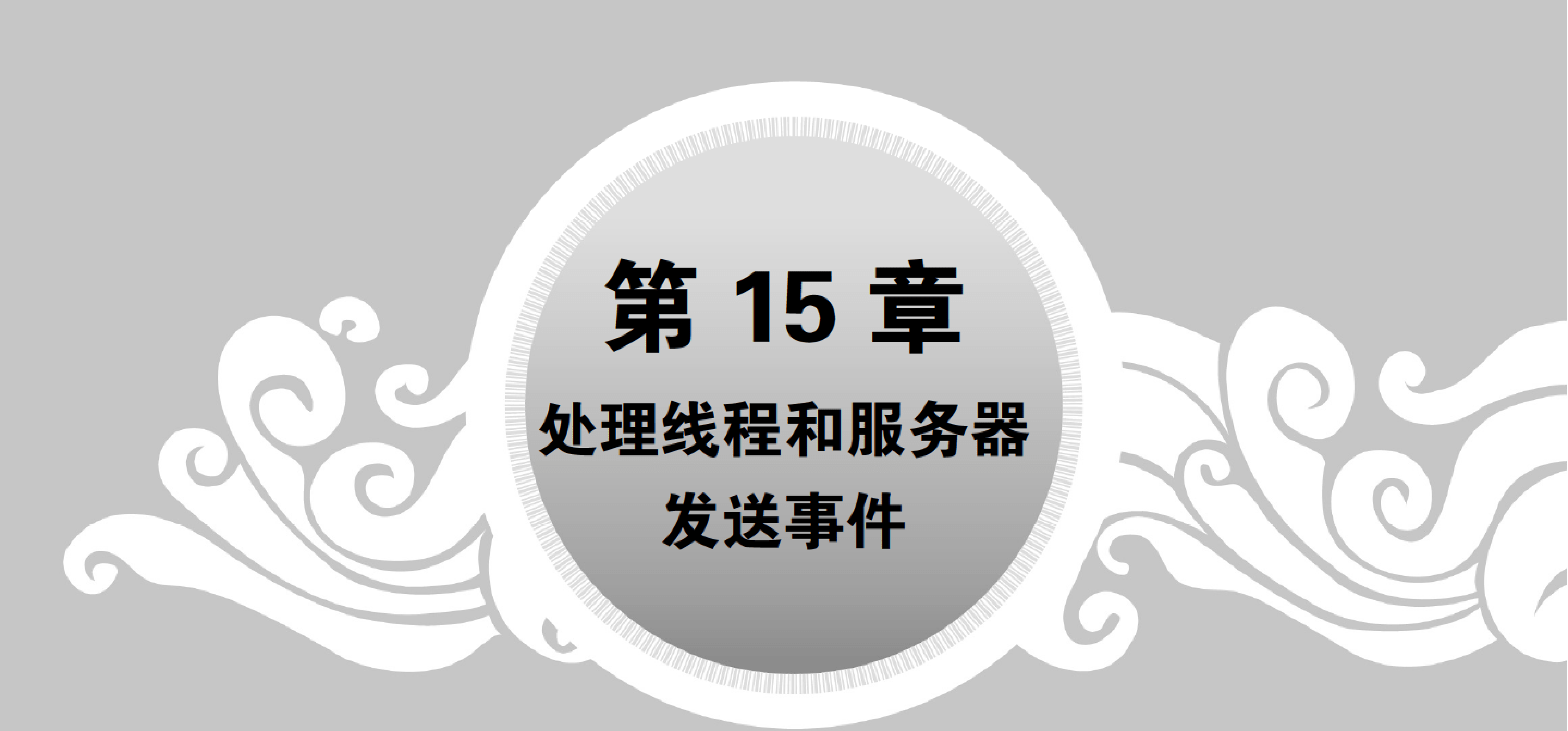
疑问 4：WebSocket 的优势在哪里？

答：它可以实现真正的实时数据通信。众所周知，B/S 模式下应用的是 HTTP 协议，是无状态的，所以不能保持持续的连接。数据交换是通过客户端提交一个 Request 到服务器端，然后服务器端返回一个 Response 到客户端来实现的。而 WebSocket 是通过 HTTP 协议的初始握手阶段然后升级到 Web Socket 协议以支持实时数据通信。

WebSocket 可以支持服务器主动向客户端推送数据。一旦服务器和客户端通过 WebSocket 建立起连接，服务器便可以主动地向客户端推送数据，而不像普通的 Web 传输方式需要先由客户端发送 Request 才能返回数据，从而增强了服务器的能力。

WebSocket 协议设计了更为轻量级的 Header，除了首次建立连接时需要发送头部和普通 Web 连接类似的数据之外，建立 WebSocket 连接后，相互沟通的 Header 就会异常的简洁，大大减少了冗余的数据传输。

WebSocket 提供了更为强大的通信能力和更为简洁的数据传输平台，能更为方便地完成 Web 开发中的双向通信功能。



第 15 章 处理线程和服务器发送事件

利用 Web Worker 技术，可以实现网页脚本程序的多线程后台执行，并且不会影响其他脚本的执行，为大型网站的顺畅运行提供了更好的实现方法。HTML 5 规范定义了 Server-Sent Event 和 Web Socket，为浏览器变成一个 RIA(Rich Internet Applications，富互联网应用)客户端平台提供了强大的支持。使用这两个特性，可以帮助服务器将数据“推送”到客户端浏览器。



15.1　Web Worker

在 HTML 5 中为了提供更好的后台程序执行，设计了 Web Worker 技术。Web Worker 的产生主要是考虑到在 HTML 4 中执行的 JavaScript Web 程序都是以单线程的方式执行的，一旦前面的脚本花费时间过长，后面的程序就会因长时间得不到响应而使用户页面操作出现异常。

15.1.1　Web Worker 概述

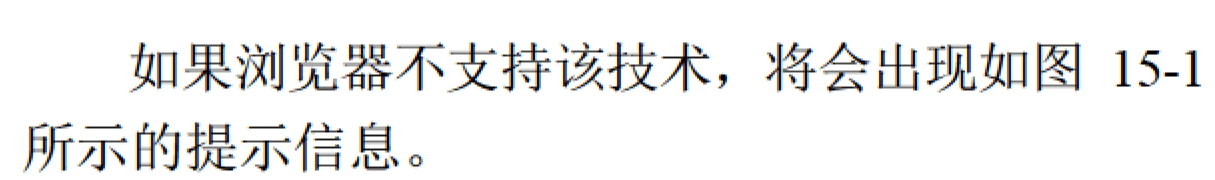
Web Worker 实现的是线程技术，可以使运行在后台的 JavaScript 独立于其他脚本，不会影响页面的性能。

Web Worker 创建后台线程的方法非常简单，只需要将在后台线程中执行的脚本文件，以 URL 地址的方式创建在 Worker 类的构造器中就可以了。其代码格式如下：

```
var worker=new worker("worker.js");
```

目前，大部分主流的浏览器都支持 Web Worker 技术。创建 Web Worker 之前，用户可以检测浏览器是否支持它，可以使用以下方法检测浏览器对 Web Worker 的支持情况：

```
if(typeof(Worker)!=="undefined")
  {
  // Yes! Web worker support!
  // Some code.....
  }
else
  {
  // Sorry! No Web Worker support..
  }
```



如果浏览器不支持该技术，将会出现如图 15-1 所示的提示信息。

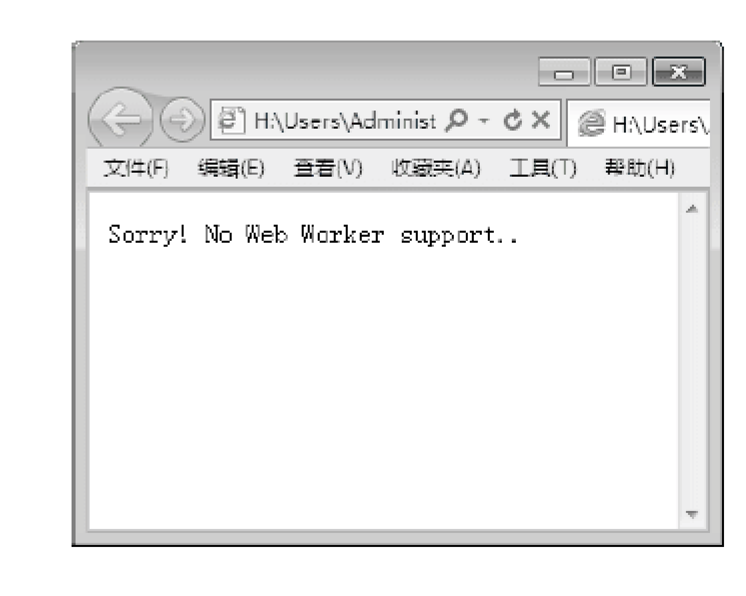


图 15-1　不支持 Web Worker 技术的提示信息

15.1.2　线程中常用的变量、函数与类

在进行 Web Worker 线程创建时会涉及一些变量、函数与类内容。其中在线程中执行的 JavaScript 脚本文件中可以用到的变量、函数与类介绍如下。

- Self：Self 关键词用来表示本线程范围内的作用域。
- Imports：导入的脚本文件必须与使用该线程文件的页面在同一个域中，并且在同一个端口中。
- ImportScripts(urls)：导入其他 JavaScript 脚本文件。参数为该脚本文件的 URL 地址，可以导入多个脚本文件。
- Onmessage：获取接收消息的事件句柄。
- Navigator 对象：与 window.navigator 对象类似，具有 appName、platform、

userAgent、appVersion 这些属性。

- setTimeout()/setInterval()：可以在线程中实现定时处理。
- XMLHttpRequest：可以在线程中处理 Ajax 请求。
- Web Workers：可以在线程中嵌套线程。
- SessionStorage/localStorage：可以在线程中使用 Web Storage。
- Close：可以结束本线程。
- Eval()、isNaN()、escape()等：可以使用所有 JavaScript 核心函数。
- Object：可以创建和使用本地对象。
- WebSockets：可以使用 WebSockets API 来向服务器发送和接收信息。
- postMessage(message)：向创建线程的源窗口发送消息。

15.1.3 案例 1——与线程进行数据的交互

在后台执行的线程是不可以访问页面和窗口对象的，但这并不妨碍前台和后台线程进行数据的交互。下面介绍一个前台和后台线程交互的案例。

在案例中，后台执行的 JavaScript 脚本线程是从 0～200 的整数中随机挑选一些整数，然后再在选出的这些整数中选择可以被 5 整除的整数，最后将这些选出的整数交给前台显示，以实现前台与后台线程的数据交互。

step 01 完成前台的网页 15.1.html，其代码内容如下：

```
<!DOCTYPE html>
<html>
<head>
<title>前台与后台线程的数据交互</title>
<script type="text/javascript">
var intArray=new Array(200);    //随机数组
var intStr="";                  //将随机数组用字符串进行连接
//生成 200 个随机数
for(var i=0;i<200;i++)
{
   intArray[i]=parseInt(Math.random()*200);
   if(i!=0)
      intStr+=";";       //用分号做随机数组的分隔符
   intStr+=intArray[i];
}
//向后台线程提交随机数组
var worker = new Worker("15.1.js");
worker.postMessage(intStr);
// 从线程中取得计算结果
worker.onmessage = function(event) {
   if(event.data!="")
   {
      var h;          //行号
      var l;          //列号
      var tr;
      var td;
      var intArray=event.data.split(";");
      var table=document.getElementById("table");
```

```
        for(var i=0;i<intArray.length;i++)
        {
            h=parseInt(i/15,0);
            l=i%15;
            //该行不存在
            if(l==0)
            {
                //添加新行的判断
                tr=document.createElement("tr");
                tr.id="tr"+h;
                table.appendChild(tr);
            }
            //该行已存在
            else
            {
                //获取该行
                tr=document.getElementById("tr"+h);
            }
            //添加列
            td=document.createElement("td");
            tr.appendChild(td);
            //设置该列数字内容
            td.innerHTML=intArray[h*15+l];
            //设置该列对象的背景色
            td.style.backgroundColor="#f56848";
            //设置该列对象数字的颜色
            td.style.color="#000000";
            //设置对象数字的宽度
            td.width="30";
        }
    }
};
</script>
</head>
<body>
<h2 style="text-shadow:0.1em 3px 6px blue">从随机生成的数字中抽取 5 的倍数并显示示
例</h2>
<table id="table">
</table>
</body>
</html>
```

step 02 为了实现后台线程，需要编写后台执行的 JavaScript 脚本文件 15.1.js，其代码如下：

```
onmessage = function(event) {
    var data = event.data;
    var returnStr;                          //将 5 的倍数组成字符串并返回
    var intArray=data.split(";");           //设置返回字符串中数字分隔符为“;”号
    returnStr="";
    for(var i=0;i<intArray.length;i++)
    {
        if(parseInt(intArray[i])%5==0)      //判断能否被 5 整除
        {
```

```
            if(returnStr!="")
                returnStr+=";";
            returnStr+=intArray[i];
        }
    }
    postMessage(returnStr);                         //返回 5 的倍数组成的字符串
}
```

step 03 使用 IE 11.0 打开编辑好的网页文件，效果如图 15-2 所示。



图 15-2　从随机生成的数字中抽取 5 的倍数并显示示例



由于数字是随机产生的，所以每次生成的数据序列都是不同的。

15.2　线 程 嵌 套

线程中可以嵌套子线程，这样就可以将后台中较大的线程切割成多个子线程，每个子线程独立完成一份工作，可以提高程序的效率。有关线程嵌套的内容介绍如下。

15.2.1　案例 2——单线程嵌套

最简单的线程嵌套是单层的嵌套。下面介绍一个单线程的嵌套案例，该案例所实现的效果和上节中案例的效果相似。其具体操作步骤如下。

step 01 完成网页前台页面 15.2.html。其具体代码如下：

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
var worker = new Worker("15.2.js");
worker.postMessage("");
// 从线程中取得计算结果
worker.onmessage = function(event) {
    if(event.data!="")
    {
        var j;    //行号
```

```
        var k;   //列号
        var tr;
        var td;
        var intArray=event.data.split(";");
        var table=document.getElementById("table");
        for(var i=0;i<intArray.length;i++)
        {
            j=parseInt(i/10,0);
            k=i%10;
            if(k==0)      //该行不存在
            {
                //添加行
                tr=document.createElement("tr");
                tr.id="tr"+j;
                table.appendChild(tr);
            }
            else  //该行已存在
            {
                //获取该行
                tr=document.getElementById("tr"+j);
            }
            //添加列
            td=document.createElement("td");
            tr.appendChild(td);
            //设置该列内容
            td.innerHTML=intArray[j*10+k];
            //设置该列背景色
            td.style.backgroundColor="blue";
            //设置该列字体颜色
            td.style.color="white";
            //设置列宽
            td.width="30";
        }
    }
};
</script>
</head>
<body>
<h2 style="text-shadow:0.1em 3px 6px blue">从随机生成的数字中抽取 5 的倍数并显示示
例</h2>
<table id="table">
</table>
</body>
</html>
```

step 02 下面需要编写程序后台执行的主线程的代码内容。该线程用于执行数据挑选，会在 0～200 随机产生 200 个随机整数(数字可重复)，并将其交给子线程，让子线程挑选可以被 5 整除的数字。代码如下：

```
onmessage=function(event){
    var intArray=new Array(200);        //产生随机的数组
    //生成 200 个随机数
    for(var i=0;i<200;i++)              //数字范围 0～200
        intArray[i]=parseInt(Math.random()*200);
```

```
    var worker;
    //调用子线程
    worker=new Worker("15.2-2.js");
    //将随机数组提交给子线程
    worker.postMessage(JSON.stringify(intArray));
    worker.onmessage = function(event) {
        //将挑选结果返回主页面
        postMessage(event.data);
    }
}
```

step 03 经过上一步主线程的数字挑选后，可以通过以下子线程将这些数字拼接成字符串，并返回主线程，其操作代码如下：

```
onmessage = function(event) {
    var intArray= JSON.parse(event.data);
    var returnStr;
    returnStr="";
    for(var i=0;i<intArray.length;i++)
    {
        //判断数字能否被 5 整除
        if(parseInt(intArray[i])%5==0)
        {
            if(returnStr!="")
                returnStr+=";";
            //将所有可以被 5 整除的数字拼接成字符串
            returnStr+=intArray[i];
        }
    }
    //返回拼接后的字符串至主线程
    postMessage(returnStr);
    //关闭子线程
    close();
}
```

step 04 使用 IE 11.0 查看网页前台页面，随机产生了一些可以被 5 整除的数字，如图 15-3 所示。

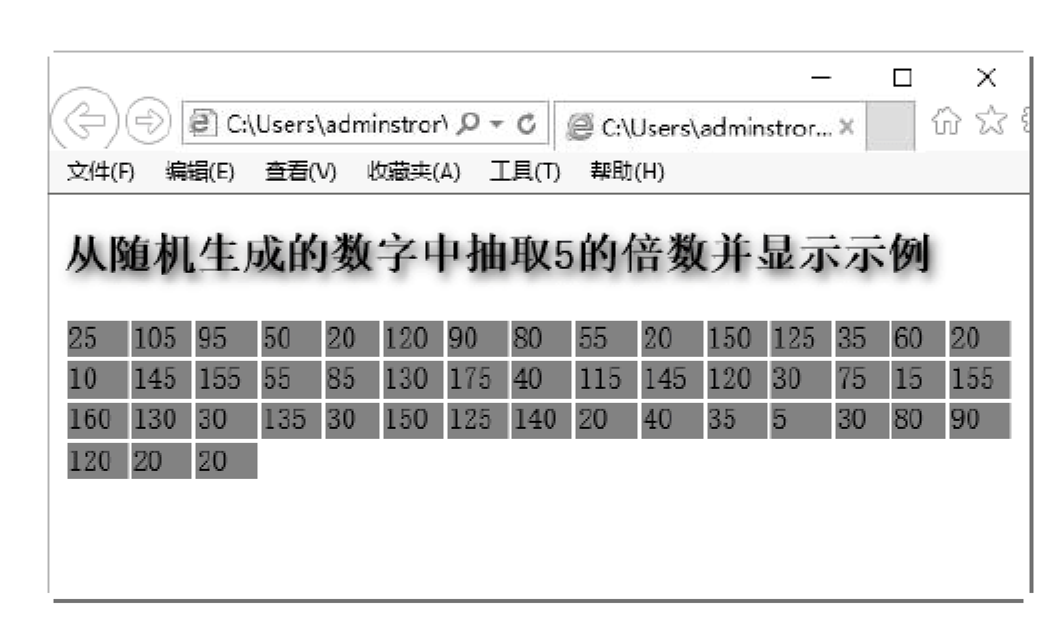


图 15-3 从随机生成的数字中抽取 5 的倍数并显示示例

15.2.2 案例 3——多个子线程中的数据交互

在实现上述案例时，也可以将子线程再次拆分，生成多个子线程，由多个子线程同时完

成工作，这样可以提高处理速度，对较大的 JavaScript 脚本程序来说很实用。

下面将上述案例的程序改为多个子线程嵌套的数据交互案例。

step 01 网页前台文件不需要修改，主线程的脚本文件 15.3.js 的内容如下：

```
onmessage=function(event){
   var worker;
   //调用发送数据的子线程
   worker=new Worker("15.3-2.js");
   worker.postMessage("");
   worker.onmessage = function(event) {
       //接收子线程中数据，本示例中为创建好的随机数组
       var data=event.data;
       //创建接收数据子线程
       worker=new Worker("15.2-2.js");
      //把从发送数据子线程中发回消息传递给接收数据的子线程
       worker.postMessage(data);
 worker.onmessage = function(event) {
             //获取接收数据子线程中传回数据，本示例中为挑选结果
   var data=event.data;
            //把挑选结果发送回主页面
            postMessage(data);
}
    }
}
```

上述代码的主线程脚本中提到了两个子线程脚本，其中一个 15.3-2.js 负责创建随机数组，并发送给主线程；另一个 15.2-2.js 负责从主线程接收选好的数组，并进行处理。15.2-2.js 脚本沿用上节脚本文件。

step 02 15.3-2.js 脚本文件的详细代码如下：

```
onmessage = function(event) {
var intArray=new Array(200);
for(var i=0;i<200;i++)
   intArray[i]=parseInt(Math.random()*200);
postMessage(JSON.stringify(intArray));
close();
}
```

step 03 执行后的效果如图 15-4 所示。

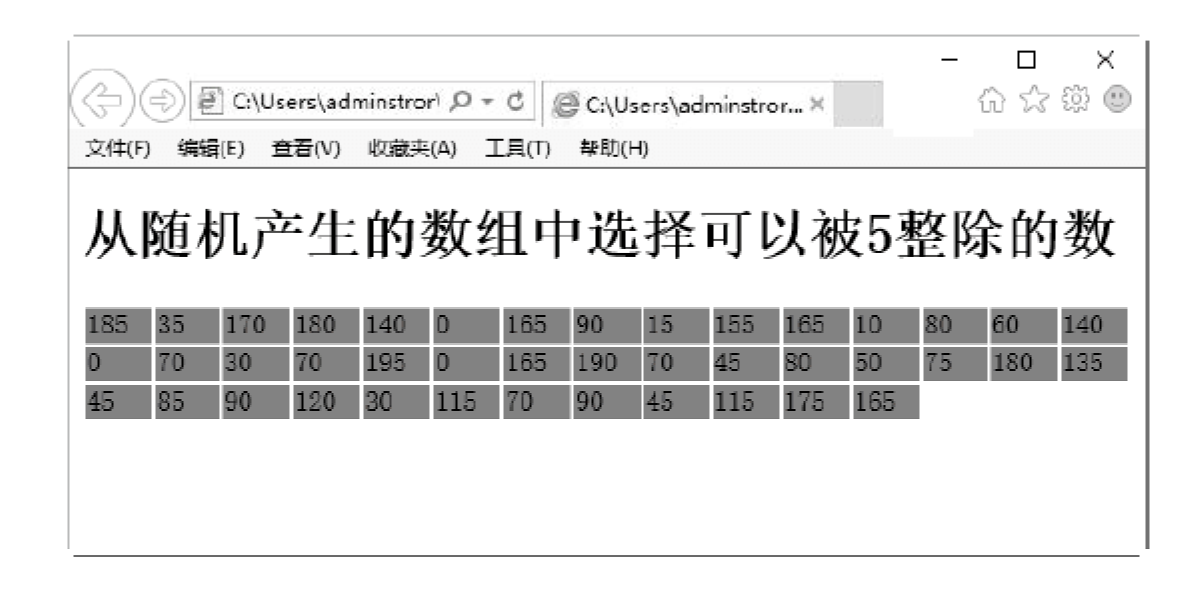


图 15-4　从随机产生的数组中选择可以被 5 整除的数



通过以上几个案例的展示，其最终显示结果都是相同的，只是代码的编辑与线程的嵌套有所差异。在实际的应用中合理地嵌套子线程，虽然代码结构会变得复杂，但是能很大程度地提高程序的处理效率。

15.3 服务器发送事件概述

在网页客户端更新过程中，如果使用早期技术，网页不得不询问是否有可用的更新，这样将不能很好地实时获取服务器的信息，并且加大了资源的耗费。在 HTML 5 中，通过服务器发送事件，可以让网页客户端自动获取来自服务器的更新。

服务器发送事件(Server-Sent Event)允许网页获得来自服务器的更新，这种数据的传递和前面章节讲述的 Web Socket 不同。服务器发送事件是单向传递信息，服务器将更新的信息自动发送到客户端，而 Web Socket 是双向通信技术。

目前，常见浏览器对 Server-Sent Event 的支持情况如表 15-1 所示。

表 15-1 常见浏览器对 Server-Sent Event 的支持情况

浏览器名称	支持 Server-Sent Event 的版本
Internet Explorer	不支持
Firefox	Firefox 3.6 及更高版本
Opera	Opera 12.0 及更高版本
Safari	Safari 5 及更高版本
Chrome	Chrome 5 及更高版本

15.4 服务器发送事件的实现过程

了解完服务器发送事件的基本概念后，下面学习其实现过程。

15.4.1 案例 4——检测浏览器是否支持 Server-Sent 事件

首先可以检查客户端浏览器是否支持 Server-Sent 事件。其代码如下：

```
if(typeof(EventSource)!=="undefined")
  {
  // 浏览器支持的情况
   }
else
  {
  // 对不起，您的浏览器不支持……
  }
```

用户在代码中设置提示信息，这样如果浏览者的客户端不支持，将会显示提示信息。

15.4.2 案例 5——使用 EventSource 对象

在 HTML 5 的服务器发送事件中，使用 EventSource 对象接收服务器发送事件的通知。该对象的事件含义如表 15-2 所示。

表 15-2 EventSource 对象的事件

事件名称	含 义
onopen	当连接打开时触发该事件
onmessage	当收到信息时触发该事件
onerror	当连接关闭时触发该事件

在事件处理函数中，可以通过使用 readyState 属性检测连接状态，主要有 3 种状态，如表 15-3 所示。

表 15-3 EventSource 对象的事件状态

状态名称	值	含 义
CONNECTING	0	正在建立连接
OPEN	1	连接已经建立，正在委派事件
CLOSED	2	连接已经关闭

例如下面的代码就是使用了 onmessage 的实例：

```
var source=new EventSource("/123.php");
source.onmessage=function(event)
  {
  document.getElementById("result").innerHTML+=event.data + "<br />";
  };
```

其中，该代码创建一个新的 EventSource 对象，然后规定发送更新的页面的 URL(本例中是“/123.php”)。每接收到一次更新，就会发生 onmessage 事件。当 onmessage 事件发生时，把已接收的数据推入 id 为 result 的元素中。

15.4.3 案例 6——编写服务器端代码

为了让上面的例子可以运行，还需要能够发送数据更新的服务器(如 PHP 和 ASP)。服务器端事件流的语法非常简单，把 Content-Type 报头设置为 text/event-stream，然后就可以开始发送事件流了。

如果服务器是 PHP，则服务器的代码如下：

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');
$time = date('r');
```

```
echo "data: The server time is: {$time}\n\n";
flush();
?>
```

如果服务器是 ASP，则服务器的代码如下：

```
<%
Response.ContentType="text/event-stream"
Response.Expires=-1
Response.Write("data: " & now())
Response.Flush()
%>
```

在上述代码中，把报头 Content-Type 设置为 text/event-stream，规定不对页面进行缓存，输出发送日期(始终以“data:”开头)，向网页刷新输出数据。

15.5　综合案例——创建 Web Worker 计数器

本实例主要创建一个简单的 Web Worker，实现在后台计数的功能。具体操作步骤如下。

step 01 首先创建一个外部的 JavaScript 文件“workers01.js”，主要用于计数，代码如下：

```
var i=0;

function timedCount()
{
i=i+1;
postMessage(i);
setTimeout("timedCount()",500);
}
timedCount();
```

以上代码中重要的部分是 postMessage()方法，主要用于向 HTML 页面传回一段消息。

step 02 创建 HTML 页面的代码如下：

```
<!DOCTYPE html>
<html>
<body>
<p>计数: <output id="result"></output></p>
<button onclick="startWorker()">开始 Worker</button>
<button onclick="stopWorker()">停止 Worker</button>
<br /><br />
<script>
var w;
function startWorker()
{
<!—首先判断浏览器是否支持 web worker -->
 if(typeof(Worker)!=="undefined")
   {
<!—检测是否存在 worker，如果不存在，它会创建一个新的 Web Worker 对象，然后运行
" workers.js01" 中的代码-->
   if(typeof(w)=="undefined")
   {
   w=new Worker("/workers01.js");
   }
```

```
<!—向 web worker 添加一个"onmessage"事件监听器-->
   w.onmessage = function (event) {
    document.getElementById("result").innerHTML=event.data;
    };
  }
else
  {
  document.getElementById("result").innerHTML="对不起，您的浏览器不支持 Web
Workers...";
  }
}
  function stopWorker()
{
<!—终止 web worker，并释放浏览器/计算机资源-->
  w.terminate();
}
</script>
</body>
</html>
```

step 03 运行结果如图 15-5 所示。

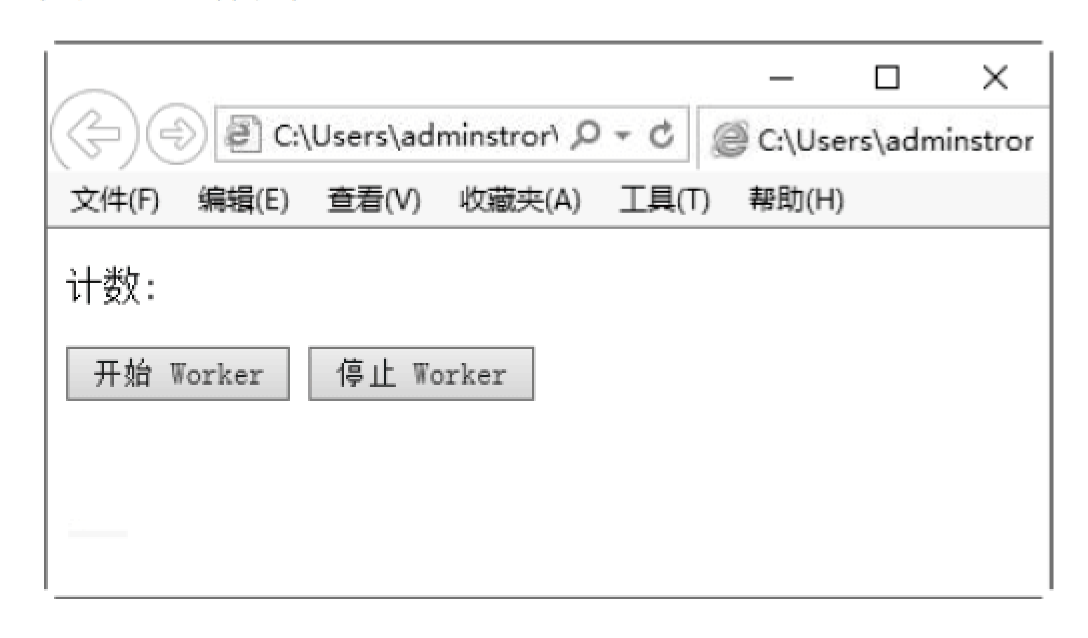


图 15-5　创建 Web Worker 计数器

15.6　跟我学上机——服务器发送事件实战应用

下面通过一个综合的案例，详细介绍服务器发送事件的操作过程。

step 01 首先创建运行主页文件，代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset=\"UTF-8\">
</head>
<body>
<h1>获得服务器更新</h1>
<div id="result">
</div>
<script>
if(typeof(EventSource)!=="undefined")
  {
  var source=new EventSource("/123.php");
  source.onmessage=function(event)
    {
```

```
    document.getElementById("result").innerHTML+=event.data + "<br />";
    };
  }
else
  {
  document.getElementById("result").innerHTML="对不起，您的浏览器不支持服务器发送事件...";
  }
</script>
</body>
</html>
```



通信数据的编码这里规定为 UTF-8 格式，另外所有的页面编码要统一为 UTF-8 格式，否则会乱码或无数据。

step 02 编写服务器端文件 123.php，代码如下：

```
<?php
error reporting(E ALL);
//注意： 发送包头定义 MIMIE 类型(header 部分)，是实现服务器所必需的代码(MIMIE 类型定义了
//事件框架格式)
header(\"Content-Type:text/event-stream\");
echo 'data:服务器第一次发送数据'.\"\n\";
echo 'data:服务器第二次发送数据'.\"\n\";
?>
```



输出的格式必须为 data:value 格式，这是 text/event-tream 格式规定。

step 03 在 IE 11.0 中访问主页文件，效果如图 15-6 所示。

step 04 在 Firefox 53.0 中访问主页文件，效果如图 15-7 所示。服务器每隔一段时间推送一个此数据。



图 15-6　在 IE 中的访问主页文件效果



图 15-7　在 Firefox 中的访问主页文件效果

15.7 高手解惑

疑问 1：工作线程(Web Worker)的主要应用场景有哪些？

答：工作线程的主要应用场景有 3 个，分别如下。

(1) 使用工作线程做后台数值(算法)计算。

(2) 使用工作线程处理多用户并发连接。

(3) HTML 5 线程代理。

疑问 2：目前浏览器对 Web Worker 的支持情况如何？

答：目前大部分主流的浏览器都支持 Web Worker，但是 Internet Explorer 9 之前的版本并不支持。

疑问 3：如何编写 JSP 的服务器端代码？

答：如果服务器端是 JSP，服务器的代码段如下：

```
<%@ page contentType="text/event-stream; charset=UTF-8"%>
<%
   response.setHeader("Cache-Control", "no-cache");
   out.print("data: >> server Time" + new java.util.Date() );
   out.flush();
%>
```

其中，编码要采用统一的 UTF-8 格式。

疑问 4：如何优化服务器端代码？

答：EventSource 对象是一个不间歇运行的程序，时间一长会大量地消耗资源，甚至导致客户端浏览器崩溃，那么如何优化执行代码呢？

在 HTML 5 中使用 Web Workers 优化 JavaScript 执行复杂运算、重复运算和多线程；对于执行时间长、消耗内存多的 JavaScript 程序代码最为有用。

第 16 章 构建离线的 Web 应用

网页离线应用程序是实现离线 Web 应用的重要技术，目前已有的离线 Web 应用程序很多。通过本章的学习，读者能够掌握 HTML 5 离线应用程序的基础知识，了解离线应用程序的实现方法。

重点案例效果

C:\Users\adminstror\
创建构成界面的H
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

Web 离线应用

check for updates

log

16.1 HTML 5 离线 Web 应用概述

在 HTML 5 中，新增了本地缓存，也就是 HTML 离线 Web 应用，主要是通过应用程序缓存整个离线网站的 HTML、CSS、JavaScript、网站图像和资源。当服务器没有和 Internet 建立连接的时候，也可以利用本地缓存中的资源文件来正常运行 Web 应用程序。

另外，如果网站发生了变化，应用程序缓存将重新加载变化的数据文件。

浏览器网页缓存与本地缓存的主要区别如下。

(1) 浏览器网页缓存主要是为了加快网页加载的速度，所以会对每一个打开的网页都进行缓存操作。而本地缓存是为整个 Web 应用程序服务的，只缓存那些指定缓存的网页。

(2) 在网络连接的情况下，浏览器网页缓存一个页面的所有文件，但是一旦离线，用户单击链接时，将会得到一个错误消息。而本地缓存在离线时，仍然可以正常访问。

(3) 对网页浏览者而言，浏览器网页缓存了哪些内容和资源，这些内容是否安全可靠等都不知道。而本地缓存的页面是编程人员指定的内容，所以在安全方面相对可靠了许多。

16.2 案例 1——使用 HTML 5 离线 Web 应用 API

离线 Web 应用较为普遍。下面详细介绍离线 Web 应用的构成与实现方法。

16.2.1 检查浏览器的支持情况

不同的浏览器版本对 Web 离线应用技术的支持情况是不同的。如表 16-1 所示是常见浏览器对 Web 离线应用的支持情况。

表 16-1 浏览器对 Web 离线应用的支持情况

浏览器名称	支持 Web 离线应用的版本情况
Internet Explorer	Internet Explorer 9 及更低版本目前尚不支持
Firefox	Firefox 3.5 及更高版本
Opera	Opera 10.6 及更高版本
Safari	Safari 4 及更高版本
Chrome	Chrome 5 及更高版本
Android	Android 2.0 及更高版本

使用离线 Web 应用 API 前最好先检查浏览器是否支持它。检查浏览器是否支持的代码如下：

```
if(windows.applicationcache){
//浏览器支持离线应用}
```



16.2.2 搭建简单的离线应用程序

为了使一个包含 HTML 文档、CSS 样式表和 javascript 脚本文件的单页面应用程序支持离线应用，需要在 HTML 5 元素中加入 manifest 特性。具体实现代码如下：

```
<!doctype html>
<html manifest="163.manifest">

</html>
```

执行以上代码可以提供一个存储的缓存空间，但是还不能完成离线应用程序的使用。需要指明哪些资源可以享用这些缓存空间，即需要提供一个缓冲清单文件。具体实现代码如下：

```
CHCHE MANIFEST
index.html
163.js
163.css
163.gif
```

以上代码中指明了 4 种类型的资源对象文件构成缓冲清单。

16.2.3 支持离线行为

要支持离线行为，首先要能够判断网络连接状态，在 HTML 5 中引入了一些判断应用程序网络连接是否正常的新的事件。对应应用程序的在线状态和离线状态会有不同的行为模式。

用于实现在线状态监测的是 window.navigator 对象的属性。其中的 navigator.online 属性是一个标明浏览器是否处于在线状态的布尔属性。当 online 值为 true 并不能保证 Web 应用程序在用户的机器上一定能访问到相应的服务器；而当其值为 false 时，不管浏览器是否真正联网，应用程序都不会尝试进行网络连接。

监测页面状态是在线还是离线的具体代码如下：

```
//页面加载的时候，设置状态为 online 或 offline
Function loaddemo(){
  If (navigator.online) {
    Log("online");
} else {
  Log("offline");
}
}
//添加事件监听器，在线状态发生变化时，触发相应动作
Window.addeventlistener("online",function{
}, true);

Window.addeventlistener("offline",function(e) {
  Log("offline");
},true);
```



上述代码可以在 Internet Explorer 中使用。

16.2.4 Manifest 文件

客户端的浏览器是如何知道应该缓存哪些文件呢？这就需要依靠 Manifest 文件来管理。Manifest 文件是一个简单文本文件，在该文件中以清单的形式列举了需要被缓存或不需要被缓存的资源文件的文件名称以及这些资源文件的访问路径。

Manifest 文件把指定的资源文件类型分为 3 类，分别是 CACHE、NETWORK 和 FALLBACK。这 3 类的含义分别如下。

(1) CACHE 类别。该类别指定需要被缓存在本地的资源文件。这里需要特别注意的是：如果为某个页面指定需要本地缓存的资源文件时，不需要把这个页面本身指定在 CACHE 类型中，因为如果一个页面具有 manifest 文件，浏览器会自动对这个页面进行本地缓存。

(2) NETWORK 类别。该类别为不进行本地缓存的资源文件，这些资源文件只有当客户端与服务器端建立连接的时候才能访问。

(3) FALLBACK 类别。该类别中指定两个资源文件，其中一个资源文件为能够在线访问时使用的资源文件；另一个资源文件为不能在线访问时使用的备用资源文件。

以下是一个简单的 manifest 文件的内容。

```
CACHE MANIFEST
#文件的开头必须是 CACHE MANIFEST
CACHE:
163.html
myphoto.jpg
16.php
NETWORK:
http://www.baidu.com/xxx
feifei.php
FALLBACK:
online.js locale.js
```

上述代码含义分析如下。

(1) 指定资源文件，文件路径可以是相对路径，也可以是绝对路径。指定时每个资源文件为独立的一行。

(2) 第一行必须是 CACHE MANIFEST，此行的作用告诉浏览器需要对本地缓存中的资源文件进行具体设置。

(3) 每一个类型都必须出现，而且同一个类别可以重复出现。如果文件开头没有指定类别而直接书写资源文件的时候，浏览器把这些资源文件视为 CACHE 类别。

(4) 在 manifest 文件中，注释行以“#”开始，主要用于进行一些必要的说明或解释。

为单个网页添加 manifest 文件时，需要在 Web 应用程序页面上的 html 元素的 manifest 属性中指定 manifest 文件的 URL 地址。具体的代码如下：

```
<html manifest="163.manifest">
</html>
```

添加上述代码后，浏览器就能够正常地阅读该文本文件。





用户可以为每一个页面单独指定一个 manifest 文件，也可以对整个 Web 应用程序指定一个总的 manifest 文件。

上述操作完成后，即可实现资源文件缓存到本地。当要对本地缓存区的内容进行修改时，只需要修改 manifest 文件。文件被修改后，浏览器可以自动检查 manifest 文件，并自动更新本地缓存区中的内容。

16.2.5 Application Cache API

传统的 Web 程序中浏览器也会对资源文件进行 cache，但是并不是很可靠，有时起不到预期的效果。而 HTML 5 中的 Application Cache 支持离线资源的访问，为离线 Web 应用的开发提供了可能。

使用 Application Cache API 的好处有以下几点。

(1) 用户可以在离线时继续使用。

(2) 缓存到本地，节省带宽，加速用户体验的反馈。

(3) 减轻服务器的负载。

Application Cache API 是一个操作应用缓存的接口，是 Windows 对象的直接子对象 window.applicationcache。window.applicationcache 对象可以触发一系列与缓存状态相关的事件。具体事件如表 16-2 所示。

表 16-2　window.applicationcache 对象事件表

事　件	接　口	触发条件	后续事件
checking	Event	用户代理检查更新或者在第一次尝试下载 manifest 文件的时候，本事件往往是事件队列中第一个被触发的	noupdate, downloading, obsolete, error
noupdate	Event	检测出 manifest 文件没有更新	无
downloading	Event	用户代理发现更新并且正在取资源，或者第一次下载 manifest 文件列表中列举的资源	progress, error, cached, updateready
progress	ProgressEvent	用户代理正在下载资源 manifest 文件中的需要缓存的资源	progress, error, cached, updateready
cached	Event	manifest 中列举的资源已经下载完成，并且已经缓存	无
updateready	Event	manifest 中列举的文件已经重新下载并更新成功，接下来 JS 可以使用 swapCache() 方法更新到应用程序中	无
obsolete	Event	manifest 的请求出现 404 或者 410 错误，应用程序缓存被取消	无

此外，没有可用更新或者发生错误时，还有一些表示更新状态的事件，具体如下：

```
Onerror
Onnoupdate
onprogress
```

该对象有一个数值型属性 window.applicationcache.status，代表了缓存的状态。缓存状态共有 6 种，如表 16-3 所示。

表 16-3　缓存的状态

数值型属性	缓存状态	含　义
0	UNCACHED	未缓存
1	IDLE	空闲
2	CHECKING	检查中
3	DOWNLOADING	下载中
4	UPDATEREADY	更新就绪
5	OBSOLETE	过期

window.applicationcache 有 3 个方法，如表 16-4 所示。

表 16-4　window.applicationcache 的方法

方 法 名	描　述
update()	发起应用程序缓存下载进程
abort()	取消正在进行的缓存下载
swapcache()	切换成本地最新的缓存环境



调用 update()方法会请求浏览器更新缓存。包括检查新版本的 manifest 文件并下载必要的新资源。如果没有缓存或者缓存已过期，则会抛出错误。

16.3　案例 2——使用 HTML 5 离线 Web 应用构建应用

下面结合上述内容的学习来构建一个离线 Web 应用程序，具体内容如下。

16.3.1　创建记录资源的 manifest 文件

首先要创建一个缓冲清单文件 163.manifest，文件中列出了应用程序需要缓存的资源。具体实现代码如下：

```
CACHE MANIFEST
# javascript
```

```
./offline.js
#./163.js
./log.js

#stylesheets
./CSS.css

#images
```

16.3.2　创建构成界面的 HTML 和 CSS

下面来实现网页结构，其中需要指明程序中用到的 javascript 文件和 CSS 文件，并且还要调用 manifest 文件。具体实现代码如下：

```
<!DOCTYPE html >
<html lang="en" manifest="163.manifest">
<head>
<title>创建构成界面的 HTML 和 CSS</title>
<script src="log.js"></script>
<script src="offline.js"></script>
<script src="163.js"></script>
<link rel="stylesheet" href="CSS.css" />
</head>

<body>
 <header>
    <h1>Web 离线应用</h1>
   </header>
   <section>
    <article>
      <button id="installbutton">check for updates</button>
      <h3>log</h3>
      <div id="info">
      </div>
      </article>
    </section>
</body>
</html>
```

注意　上述代码中有两点需要注意。其一，因为使用了 manifest 特性，所以 HTML 元素不能省略(为了使代码简洁，HTML 5 中允许省略不必要的 HTML 元素)。其二，代码中引入了按钮，其功能是允许用户手动安装 Web 应用程序，以支持离线情况。

16.3.3　创建离线的 JavaScript

在网页设计中经常会用到 JavaScript 文件，该文件通过<script>标签引入网页。在执行离线 Web 应用时，这些 JavaScript 文件也会一并存储到缓存中。

```
<offline.js>
/*
```

```
 *记录 window.applicationcache 触发的每一个事件
 */

window.applicationcache.onchecking =
function(e) {
 log("checking for application update");
    }
window.applicationcache.onupdateready =
function(e) {
 log("application update ready");
    }
window.applicationcache.onobsolete =
function(e) {
 log("application obsolete");
    }
window.applicationcache.onnoupdate =
function(e) {
 log("no application update found");
    }
window.applicationcache.oncached =
function(e) {
 log("application cached");
    }
window.applicationcache.ondownloading =
function(e) {
 log("downloading application update");
    }
window.applicationcache.onerror =
function(e) {
 log("online");
    }, true);
/*
 *将 applicationcache 状态代码转换成消息
 */
 showcachestatus = function(n) {
    statusmessages = ["uncached","idle","checking","downloading","update
        ready","obsolete"];
    return statusmessages[n];
}
install = function(){
 log("checking for updates");
    try {
     window.applicationcache.update();
     } catch (e) {
     applicationcache.onerror();
     }
  }
onload = function(e) {
 //检测所需功能的浏览器支持情况
    if(!window.applicationcache) {
     log("html5 offline applications are not supported in your browser.");
        return;
     }
     if(!window.localstorage) {
```

```
    log("html5 local storage not supported in your browser.");
      return;
    }
    if(!navigator.geolocation) {
    log("html5 geolocation is not supported in your browser.");
      return;
    }
    log("initial cache status: " + showcachestatus(window.applicationcache.
        status));
    document.getelementbyid("installbutton").onclick = checkfor;
}

<log.js>
log = function() {
 var p = document.createelement("p");
 var message = array.prototype.join.call(arguments," ");
   p.innerhtml = message
   document.getelementbyid("info").appendchild(p);
}
```

16.3.4　检查 Application Cache 的支持情况

Application Cache 对象并非所有浏览器都可以支持，所以在编辑时需要加入浏览器支持性检测功能，并提醒浏览者页面无法访问是浏览器兼容问题。具体实现代码如下：

```
onload = function(e) {
  // 检测所需功能的浏览器支持情况
  if (!window.applicationcache) {
     log("您的浏览器不支持 HTML 5 Offline Applications ");
    return;
  }
  if (!window.localStorage) {
     log("您的浏览器不支持 HTML 5 Local Storage  ");
    return;
  }
  if (!window.WebSocket) {
     log("您的浏览器不支持 HTML 5 WebSocket ");
    return;
  }
  if (!navigator.geolocation) {
     log("您的浏览器不支持 HTML 5 Geolocation ");
    return;
  }
    log("lnitial cache status:" +
showCachestatus(window.applicationcache.status));
  document.getelementbyld("installbutton").onclick = install;
}
```

16.3.5　为 Update 按钮添加处理函数

下面来设置 Update 按钮的行为函数，该函数功能为执行更新应用缓存。具体代码如下：

```
Install = function() {
 Log("checking for updates");
 Try {
    Window.applicationcache.update();
 } catch (e) {
    Applicationcache.onerror():
 }
}
```

说明：单击按钮后将检查缓存区，并更新需要更新的缓存资源。当所有可用更新都下载完毕之后，将向用户界面返回一条应用程序安装成功的提示信息，接下来用户就可以在离线模式下运行了。

16.3.6 添加 Storage 功能代码

当应用程序处于离线状态时，需要将数据更新写入本地存储。本实例使用 Storage 实现该功能，因为当上传请求失败后可以通过 Storage 得到恢复。如果应用程序遇到某种原因导致的网络错误，或者应用程序被关闭的时候，数据会被存储以便下次再进行传输。

实现 Storage 功能的具体代码如下：

```
Var storelocation =function(latitude, longitude){
//加载 localstorage 的位置列表
Var locations = json.pares(localstorage.locations || "[]");
//添加地理位置数据
Locations.push({"latitude" : latitude, "longitude" : longitude});
//保存新的位置列表
Localstorage。Locations = json.stringify(locations);
```

由于 localstorage 可以将数据存储在本地浏览器中，特别适用于具有离线功能的应用程序，所以本实例中使用了它来保存坐标。本地存储中的缓存数据在网络连接恢复正常后，应用程序会自动与远程服务器进行数据同步。

16.3.7 添加离线事件处理程序

对于离线 Web 应用程序，在使用时要结合当前状态执行特定的事件处理程序。本实例中的离线事件处理程序设计如下。

(1) 如果应用程序在线，事件处理函数会存储并上传当前坐标。

(2) 如果应用程序离线，事件处理函数只存储不上传。

(3) 当应用程序重新连接到网络后，事件处理函数会在 UI 上显示在线状态，并在后台上传之前存储的所有数据。

具体实现代码如下：

```
Window.addeventlistener("online", function(e){
   Log("online");
}, true);
Window.addeventlistener("offline", function(e) {
  Log("offline");
}, true);
```

网络连接状态在应用程序没有真正运行的时候可能会发生改变。例如，用户关闭了浏览器，刷新页面或跳转到了其他网站。为了应对这些情况，离线应用程序在每次页面加载时都会检查与服务器的连接状况。如果连接正常，会尝试与远程服务器同步数据。

```
If(navigator.online){
  Uploadlocations();
}
```

16.4 综合案例——离线定位跟踪

下面结合上述内容的学习来构建一个离线 Web 应用程序，具体内容如下。

step 01 创建记录资源的 manifest 文件。

首先要创建一个缓冲清单文件 163.manifest，文件中列出了应用程序需要缓存的资源。具体实现代码如下：

```
CACHE MANIFEST
# javascript
./offline.js
#./163.js
./log.js
#stylesheets
./CSS.css
#images
```

step 02 创建构成界面的 HTML 和 CSS。

下面来实现网页结构，其中需要指明程序中用到的 javascript 文件和 CSS 文件，并且还要调用 manifest 文件。具体实现代码如下：

```
<!DOCTYPE html >
<html lang="en" manifest="163.manifest">
<head>
<title>创建构成界面的 HTML 和 CSS</title>
<script src="log.js"></script>
<script src="offline.js"></script>
<script src="163.js"></script>
<link rel="stylesheet" href="CSS.css" />
</head>
<body>
 <header>
    <h1>Web 离线应用</h1>
    </header>
    <section>
    <article>
        <button id="installbutton">check for updates</button>
        <h3>log</h3>
        <div id="info">
        </div>
        </article>
    </section>
```

```
</body>
</html>
```

step 03 创建离线的 JavaScript。

在网页设计中经常会用到 javascript 文件，该文件通过<script>标签引入网页。在执行离线 Web 应用时，这些 javascript 文件也会一并存储到缓存中。代码如下：

```
<offline.js>
/*
 *记录 window.applicationcache 触发的每一个事件
 */
window.applicationcache.onchecking =
function(e) {
 log("checking for application update");
    }
window.applicationcache.onupdateready =
function(e) {
 log("application update ready");
    }
window.applicationcache.onobsolete =
function(e) {
 log("application obsolete");
    }
window.applicationcache.onnoupdate =
function(e) {
 log("no application update found");
    }
window.applicationcache.oncached =
function(e) {
 log("application cached");
    }
window.applicationcache.ondownloading =
function(e) {
 log("downloading application update");
    }
window.applicationcache.onerror =
function(e) {
 log("online");
    }, true);
/*
 *将 applicationcache 状态代码转换成消息
 */
 showcachestatus = function(n) {
    statusmessages = ["uncached","idle","checking","downloading","update
ready","obsolete"];
    return statusmessages[n];
}
install = function(){
 log("checking for updates");
    try {
     window.applicationcache.update();
     } catch (e) {
     applicationcache.onerror();
     }
```

```
  }
onload = function(e) {
 //检测所需功能的浏览器支持情况
   if(!window.applicationcache) {
    log("html5 offline applications are not supported in your browser.");
       return;
    }
    if(!window.localstorage) {
    log("html5 local storage not supported in your browser.");
       return;
    }
    if(!navigator.geolocation) {
    log("html5 geolocation is not supported in your browser.");
       return;
    }
    log("initial cache status: " + showcachestatus(window.applicationcache.
          status));
    document.getelementbyid("installbutton").onclick = checkfor;
}
<log.js>
log = function() {
 var p = document.createelement("p");
 var message = array.prototype.join.call(arguments," ");
   p.innerhtml = message
   document.getelementbyid("info").appendchild(p);
}
```

step 04 检查 Application Cache 的支持情况。

Application Cache 对象并非所有浏览器都可以支持，所以在编辑时需要加入浏览器支持性检测功能，并提醒浏览者页面无法访问是浏览器兼容问题。具体实现代码如下：

```
onload = function(e) {
  // 检测所需功能的浏览器支持情况
  if (!window.applicationcache) {
    log("您的浏览器不支持 HTML 5 Offline Applications ");
    return;
  }
  if (!window.localStorage) {
    log("您的浏览器不支持 HTML 5 Local Storage  ");
    return;
  }
    if (!window.WebSocket) {
    log("您的浏览器不支持 HTML 5 WebSocket ");
    return;
  }
     if (!navigator.geolocation) {
    log("您的浏览器不支持 HTML 5 Geolocation ");
    return;
  }
    log("lnitial cache status:" +
showCachestatus(window.applicationcache.status));
  document.getelementbyld("installbutton").onclick = install;
}
```

step 05 为 Update 按钮添加处理函数。

下面来设置 Update 按钮的行为函数，该函数功能为执行更新应用缓存。具体代码如下：

```
Install = function() {
Log("checking for updates");
Try {
    Window.applicationcache.update();
} catch (e) {
    Applicationcache.onerror():
}
}
```



单击按钮后将检查缓存区，并更新需要更新的缓存资源。当所有可用更新都下载完毕之后，将向用户界面返回一条应用程序安装成功的提示信息，接下来用户就可以在离线模式下运行了。

step 06 添加 Storage 功能代码。

当应用程序处于离线状态时，需要将数据更新写入本地存储。本实例使用 Storage 实现该功能，因为当上传请求失败后可以通过 Storage 得到恢复。如果应用程序遇到某种原因导致的网络错误，或者应用程序被关闭的时候，数据会被存储以便下次再进行传输。

实现 Storage 功能的具体代码如下：

```
Var storelocation =function(latitude, longitude){
//加载 localstorage 的位置列表
Var locations = json.pares(localstorage.locations || "[]");
//添加地理位置数据
Locations.push({"latitude" : latitude, "longitude" : longitude});
//保存新的位置列表
Localstorage。Locations = json.stringify(locations);
```

由于 localstorage 可以将数据存储在本地浏览器中，特别适用于具有离线功能的应用程序，所以本实例中使用了它来保存坐标。本地存储中的缓存数据在网络连接恢复正常后，应用程序会自动与远程服务器进行数据同步。

step 07 添加离线事件处理程序。

对于离线 Web 应用程序，在使用时要结合当前状态执行特定的事件处理程序。本实例中的离线事件处理程序设计如下。

(1) 如果应用程序在线，事件处理函数会存储并上传当前坐标。

(2) 如果应用程序离线，事件处理函数只存储不上传。

(3) 当应用程序重新连接到网络后，事件处理函数会在 UI 上显示在线状态，并在后台上传之前存储的所有数据。

具体实现代码如下：

```
Window.addeventlistener("online", function(e){
   Log("online");
}, true);
Window.addeventlistener("offline", function(e) {
  Log("offline");
}, true);
```

网络连接状态在应用程序没有真正运行的时候可能会发生改变。例如，用户关闭了浏览器，刷新页面或跳转到了其他网站。为了应对这些情况，离线应用程序在每次页面加载时都会检查与服务器的连接状况。如果连接正常，会尝试与远程服务器同步数据。

```
If(navigator.online){
  Uploadlocations();
}
```

step 08 在 IE 中预览，效果如图 16-1 所示。

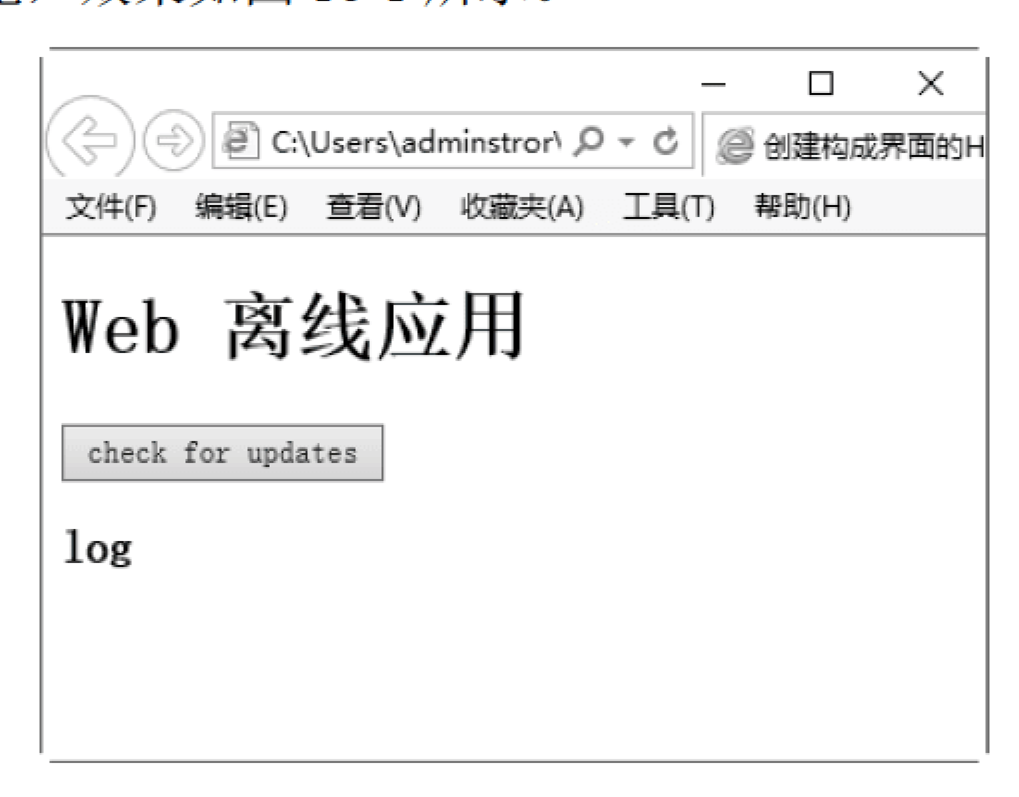


图 16-1 Web 离线应用

16.5 高 手 解 惑

疑问 1：不同的浏览器可以读取同一个 Web 中存储的数据吗？

答：在 Web 存储时，不同的浏览器将存储在不同的 Web 存储库中。例如，如果用户使用的是 IE 浏览器，那么 Web 存储工作时，将所有数据存储在 IE 的 Web 存储库中。如果用户再次使用火狐浏览器访问该站点，将不能读取 IE 浏览器存储的数据，可见每个浏览器的存储是分开并独立工作的。

疑问 2：离线存储站点时是否需要浏览者同意？

答：和地理定位类似，在网站使用 manifest 文件时，浏览器会提供一个权限提示，提示用户是否将离线设为可用，但不是每一个浏览器都支持这样的操作的。

第 IV 篇

移动开发

- 第 17 章　jQuery Mobile 基础
- 第 18 章　jQuery Mobile UI 组件
- 第 19 章　jQuery Mobile 事件
- 第 20 章　数据存储和读取技术

第 17 章 jQuery Mobile 基础

使用 HTML 5 不仅可以设计 PC 端网页，还可以开发移动网站。HTML 5 需要和函数库 jQuery Mobile 一起开发动移动网站，可以解决不同移动设备上显示界面统一的问题。本章重点学习 jQuery Mobile 的基础知识。



17.1 认识 jQuery Mobile

jQuery Mobile 是 jQuery 在手机上和平板设备上的版本。jQuery Mobile 不仅会给主流移动平台带来 jQuery 核心库，而且会发布一个完整统一的 jQuery 移动 UI 框架。通过 jQuery Mobile 制作出来的网页能够支持全球主流的移动平台，而且在浏览网页时，能够拥有操作应用软件一样的触碰和滑动效果。

jQuery Mobile 的优势如下。

(1) 简单易用。jQuery Mobile 简单易用。页面开发主要使用标记，无须或仅需很少 JavaScript。jQuery Mobile 通过 HTML 5 标记和 CSS 3 规范来配置和美化页面，对已经熟悉 HTML 5 和 CSS 3 的读者来说，掌握非常容易，架构清晰。

(2) 跨平台。目前大部分的移动设备浏览器都支持 HTML 5 标准和 jQuery Mobile，所以可以实现跨不同的移动设备。例如 Android、Apple iOS、BlackBerry、Windows Phone、Symbian 和 MeeGo 等。

(3) 提供丰富的函数库。常见的键盘、触碰功能等，开发人员不用编写代码，只需要经过简单的设置，就可以实现需要的功能，大大减少了程序员开发的时间。

(4) 丰富的布景主题和 ThemeRoller 工具。jQuery Mobile 提供了布局主题，通过这些主题，可以轻轻松松地快速创建绚丽多彩的网页。通过使用 jQuery UT 的 ThemeRoller 在线工具，只需要在下拉菜单中进行简单的设置，就可以制作出丰富多彩的网页风格，并且可以将代码下载下来应用。

jQuery Mobile 的操作流程如下。

(1) 创建 HTML 5 文件。

(2) 载入 jQuery、jQuery Mobile 和 jQuery Mobile CSS 链接库。

(3) 使用 jQuery Mobile 定义的 HTML 标准，编写网页架构和内容。

17.2 跨平台移动设备网页 jQuery Mobile

学习移动设备的网页设计开发，遇到最大的难题是跨浏览器支持的问题。为了解决这个问题，jQuery 推出了新的函数库 jQuery Mobile，主要用于统一当前移动设备的用户界面。

17.2.1 案例 1——移动设备模拟器

网页制作完成后，需要在移动设备上预览最终的效果。为了方便预览效果，用户可以使用移动设备模拟器。常见的移动设备模拟器是 Opera Mobile Emulator。

Opera Mobile Emulator 是一款针对电脑桌面开发的模拟移动设备的浏览器，几乎完全重现 Opera Mobile 手机浏览器的使用效果，可自行设置需要模拟的不同型号的手机和平板电脑配置，然后在电脑上模拟各类手机等移动设备访问网站。

Opera Mobile Emulator 的下载网址：http://www.opera.com/zh-cn/developer/mobile-

emulator/，根据不同的系统选择不同的版本，这里选择 Windows 系统下的版本，如图 17-1 所示。

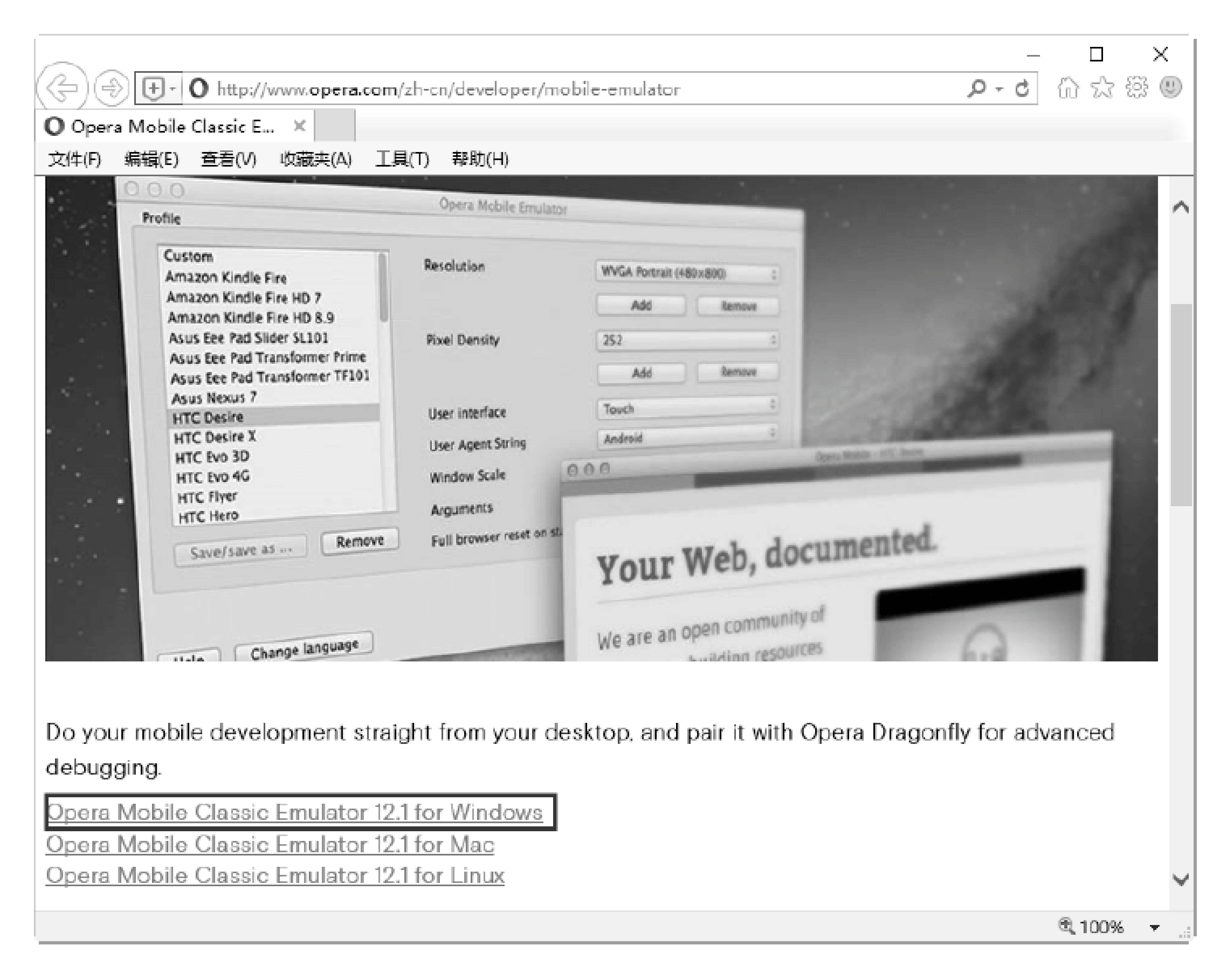


图 17-1　Opera Mobile Emulator 的下载页面

下载并安装之后启动 Opera Mobile Emulator，打开如图 17-2 所示的窗口。在“资料”列表中选择移动设备的类型，这里选择 LG Optimus 3D 选项，单击“启动”按钮。

打开欢迎界面，用户可以单击不同的链接，查看该软件的功能，如图 17-3 所示。

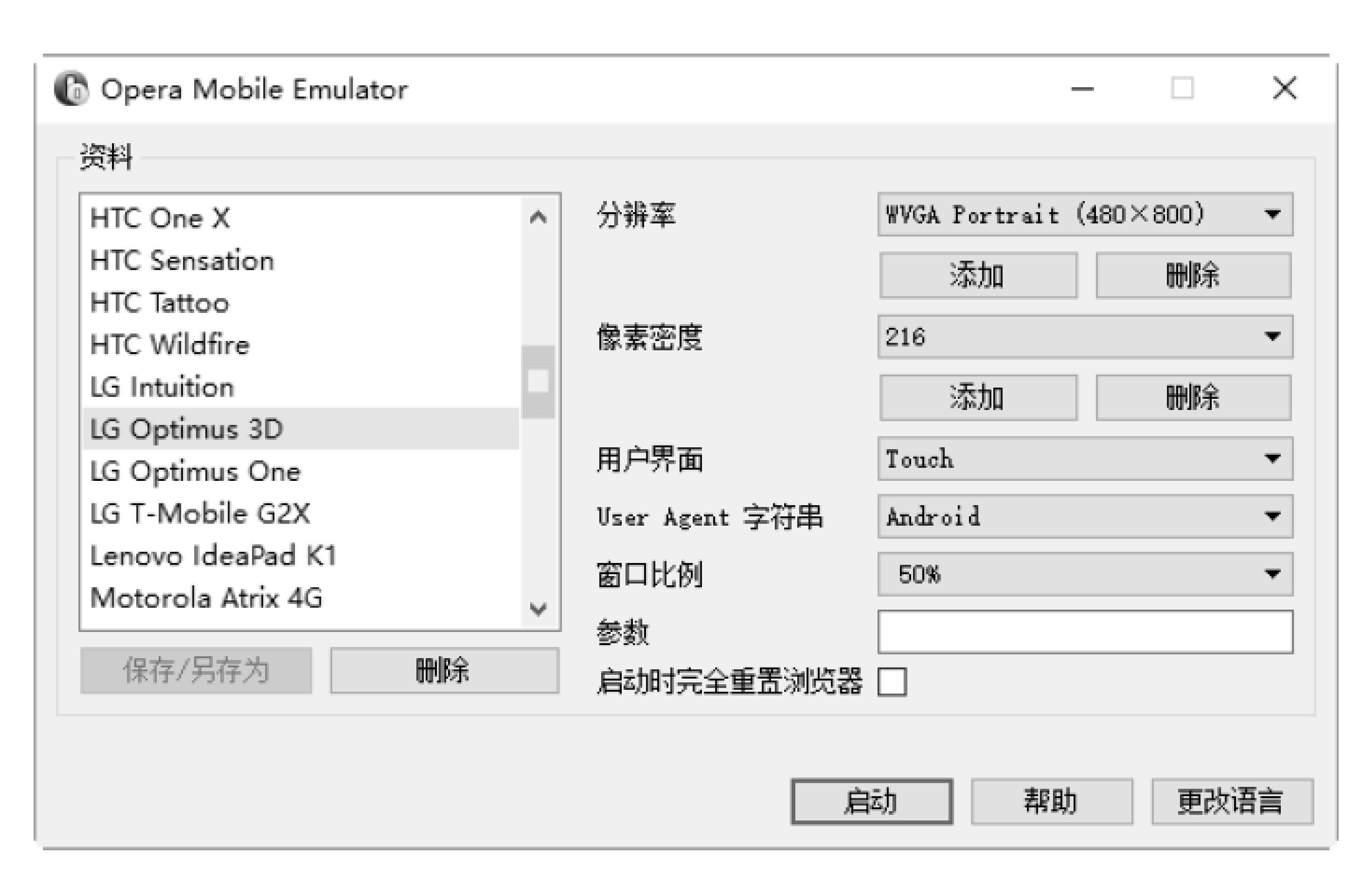


图 17-2　参数设置界面

图 17-3　欢迎界面

单击“接受”按钮，打开手机模拟器窗口，在“输入网址”文本框中输入需要查看网页效果的地址即可，如图 17-4 所示。

例如，这里直接单击“当当网”图标，即可查看当当网在该移动设备模拟器中的效果，如图 17-5 所示。



图 17-4　手机模拟器窗口



图 17-5　查看预览效果

Opera Mobile Emulator 不仅可以查看移动网页的效果，还可以任意调整窗口的大小，从而可以查看不同屏幕尺寸的效果。这一点也是 Opera Mobile Emulator 与其他移动设备模拟器相比最大的优势。

17.2.2　案例 2——jQuery Mobile 的安装

想要开发 jQuery Mobile 网页，必须要引用 JavaScript 函数库(.js)、CSS 样式表和配套的 jQuery 函数库文件。常见的引用方法有以下两种。

1. 直接引用 jQuery Mobile 库文件

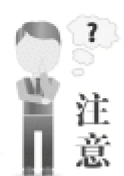
从 jQuery Mobile 的官网下载该库文件(网址是 http://jquerymobile.com/download/)，如图 17-6 所示。



图 17-6　下载 jQuery Mobile 库文件

下载完成即可解压，然后直接引用文件即可，代码如下：

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="jquery.mobile-1.4.5.css">
<script src="jquery.js"></script>
<script src="jquery.mobile-1.4.5.js"></script>
</head>
```



将下载的文件解压并和网页位于同一目录下，否则会无法引用而报错。

细心的读者会发现，在<script>标签中没有插入 type="text/javascript"，这是因为所有的浏览器中 HTML 5 的默认脚本语言就是 JavaScript，所以在 HTML 5 中已经不再需要该属性。

2. 从 CDN 中加载 jQuery Mobile

CDN 的全称是 Content Delivery Network，即内容分发网络。其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输得更快、更稳定。

使用 CDN 中加载 jQuery Mobile，用户不需要在电脑上安装任何东西。用户仅仅需要在网页中加载层叠样式(.css)和 JavaScript 库 (.js) 就能够使用 jQuery Mobile。

用户可以从 jQuery Mobile 官网中查找引用路径，网址是 http://jquerymobile.com/download/，进入该网站后，找到 jQuery Mobile 的引用链接，然后将其复制后添加到 HTML 文件<head>标记中即可，如图 17-7 所示。

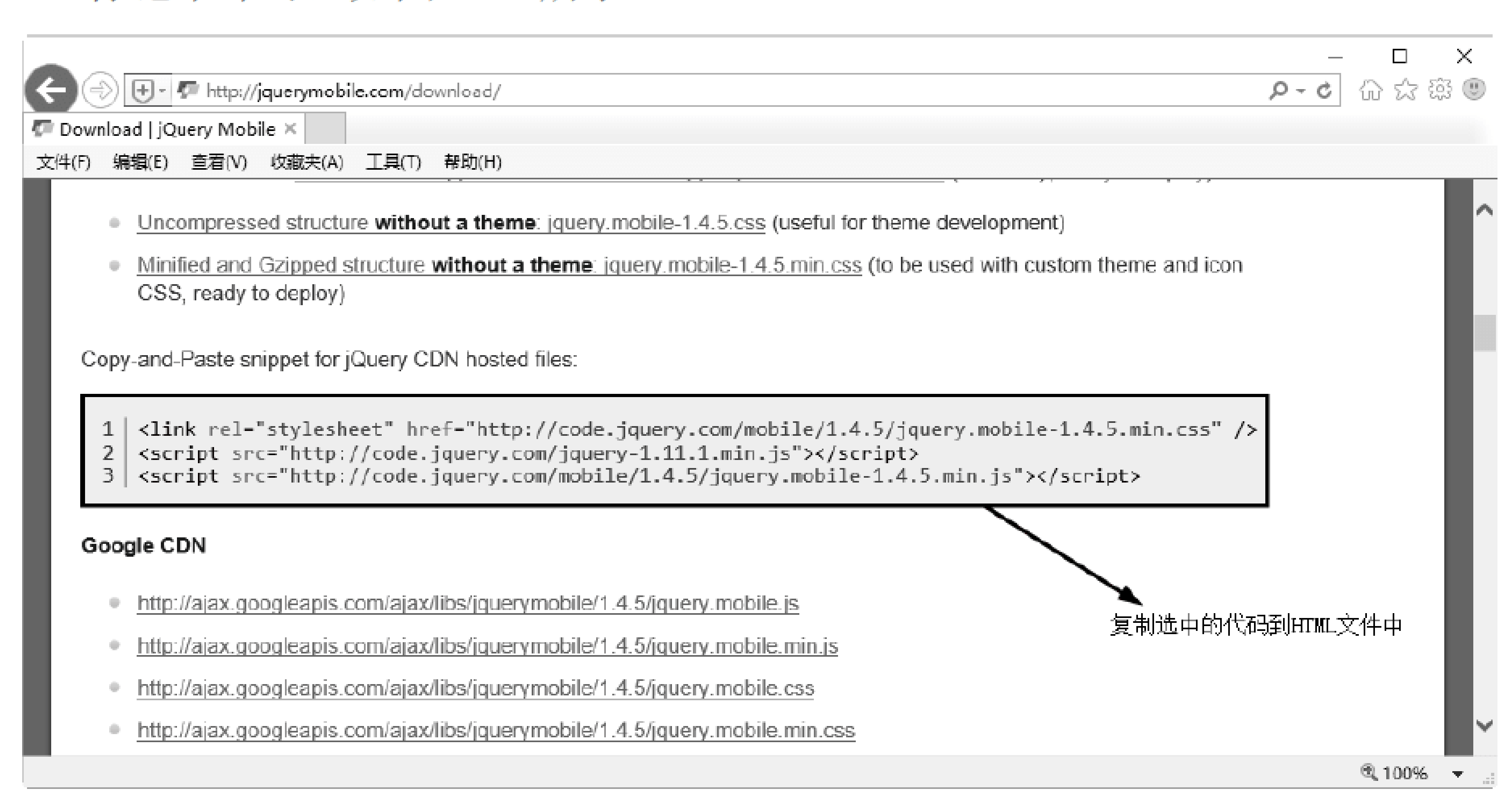


图 17-7　复制代码

将代码复制到<head>标记块内，代码如下：

```
<head>
<!-- meta 使用 viewport 以确保页面可自由缩放 -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- 引入 jQuery Mobile 样式 -->
 <link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
```

```
    <!-- 引入 jQuery 库 -->
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
    <!-- 引入 jQuery Mobile 库 -->
    <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
    </head>
```



由于 jQuery Mobile 函数库仍然在开发中，所以引用的链接中的版本号可能会与本书不同，请使用官方提供的最新版本，只要按照上述方法将代码复制下来引用即可。

17.2.3 案例 3—— jQuery Mobile 网页的架构

jQuery Mobile 网页是由 header、content 与 footer 3 个区域组成的架构，利用<div>标记加上 HTML 5 自定义属性 data-*来定义移动设备网页组件样式，最基本的属性 data-role 可以用来定义移动设备的页面架构，语法格式如下：

```
<div data-role="page">
<!—开始一个 page-->
  <div data-role="header">
    <h1>这个是标题</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>这里是内容</p>
  </div>
  <div data-role="footer">
    <h1>底部文本</h1>
  </div>
</div>
```

在模拟器中的预览效果如图 17-8 所示。

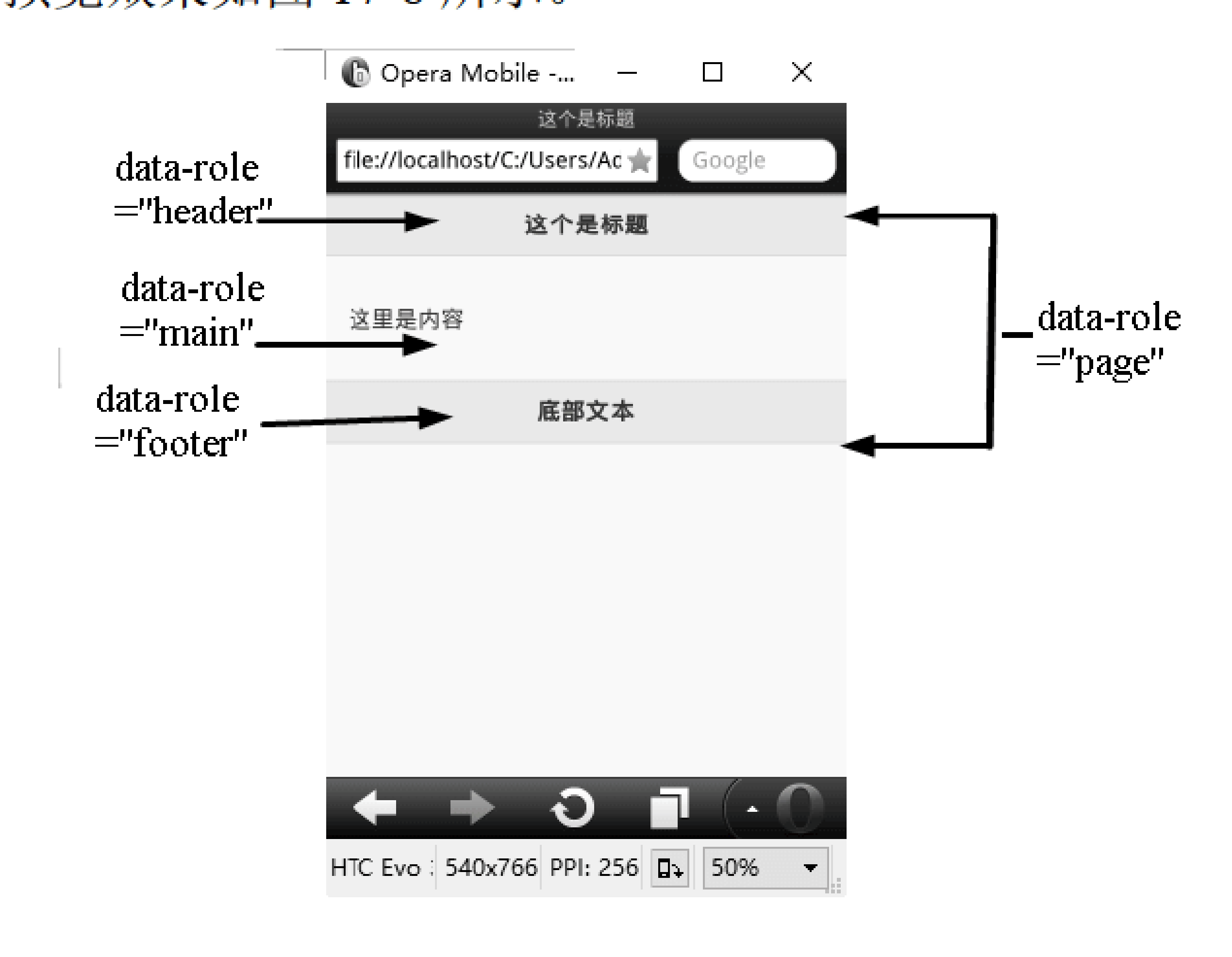


图 17-8　程序预览效果

从结果可以看出，jQuery Mobile 网页以页(page)为单位，一个 HTML 页面可以放一个页面，也可以放多个页面，只是浏览器每次只会显示一页，如果有多个页面，需要在页面中添加超链接，从而实现多个页面的切换。

代码分析如下。

(1) data-role="page" 是在浏览器中显示的页面。

(2) data-role="header" 是在页面顶部创建的工具条，通常用于标题或者搜索按钮。

(3) data-role="main" 定义了页面的内容，如文本、图片、表单、按钮等。

(4) "ui-content" 类用于在页面添加内边距和外边距。

(5) data-role="footer" 用于创建页面底部工具条。

17.3 案例 4——创建多页面的 jQuery Mobile 网页

本案例将使用 jQuery Mobile 制作一个多页面的 jQuery Mobile 网页，并创建多个页面。使用不同的 id 属性来区分不同的页面。

【例 17.1】创建多页面的 jQuery Mobile 网页(案例文件：ch17\17.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
  1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>几回花下坐吹箫，银汉红墙入望遥。</p>
    <a href="#second">下一页</a>
  </div>
  <div data-role="footer">
    <h1>清代诗人</h1>
  </div>
</div>
<div data-role="page" id="second">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>似此星辰非昨夜，为谁风露立中宵。</p>
    <a href="#first">上一页</a>
  </div>
  <div data-role="footer">
```

```
      <h1>清代诗人</h1>
    </div>
  </div>
  </body>
  </html>
```

在模拟器中的预览效果如图 17-9 所示。单击“下一页”超链接，即可进入第二页，如图 17-10 所示。单击“上一页”超链接，即可返回到第一页中。



图 17-9　程序预览效果



图 17-10　第二页预览效果

17.4　案例 5——将页面作为对话框使用

对话框是用于页面信息的显示或者表单信息的输入。jQuery Mobile 通过在链接中添加如下属性，即可将页面作为对话框使用：

```
data-dialog="true"
```

【例 17.2】 将页面作为对话框使用(案例文件：ch17\17.2.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗鉴赏</h1>
  </div>
```

```
  <div data-role="main" class="ui-content">
    <p>浩荡离愁白日斜，吟鞭东指即天涯。落红不是无情物，化作春泥更护花。</p>
    <a href="#second">查看详情</a>
  </div>
  <div data-role="footer">
    <h1>清代诗词</h1>
  </div>
</div>
<div data-role="page" data-dialog="true" id="second">
  <div data-role="header">
    <h1>诗词鉴赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>这首诗是《己亥杂诗》的第五首，写诗人离京的感受。虽然载着“浩荡离愁”，却表示仍然要
为国为民尽自己最后一份心力。</p>
    <a href="#first">上一页</a>
  </div>
  <div data-role="footer">
    <h1>清代诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 17-11 所示。单击“查看详情”超链接，即可打开一个对话框，如图 17-12 所示。



图 17-11　程序预览效果



图 17-12　对话框预览效果

从结果可以看出，对话框与普通页面不同，它显示在当前页面上，但又不会填充完整的页面，顶部图标⊗用于关闭对话框，单击“上一页”超链接，也可以关闭对话框。

17.5　案例 6——绚丽多彩的页面切换效果

jQuery Mobile 提供了各种页面切换到下一个页面的效果。主要通过设置 data-transition 属

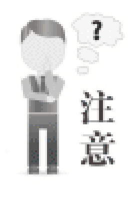
性来完成各种页面切换效果，其语法格式如下：

```
<a href="#link" data-transition="切换效果">切换下一页</a>
```

其中切换效果有很多，如表 17-1 所示。

表 17-1　页面切换效果

页面效果参数	含　义
fade	默认的切换效果。淡入到下一页
none	无过渡效果
flip	从后向前翻转到下一页
flow	抛出当前页，进入下一页
pop	像弹出窗口那样转到下一页
slide	从右向左滑动到下一页
slidefade	从右向左滑动并淡入到下一页
slideup	从下到上滑动到下一页
slidedown	从上到下滑动到下一页
turn	转向下一页



在 jQuery Mobile 的所有链接上，默认使用淡入淡出的效果。

例如，设置页面从右向左滑动到下一页，代码如下：

```
<a href="#second" data-transition="slide">切换下一页</a>
```

上面的所有效果支持后退行为。例如，用户想让页面从左向右滑动，可以添加 data-direction 属性为 reverse 值即可，代码如下：

```
<a href="#second" data-transition="slide" data-direction="reverse">切换下一
页</a>
```

【例 17.3】 实现不同的页面切换效果(案例文件：ch17\17.3.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
```

```
  <div data-role="main" class="ui-content">
    <p>老农家贫在山住，耕种山田三四亩。</p>
<!—实现从右到左切换到下一页 -->
    <a href="#second" data-transition="slide" >下一页</a>
  </div>
  <div data-role="footer">
    <h1>野老歌</h1>
  </div>
</div>
<div data-role="page" id="second">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>岁暮锄犁傍空室，呼儿登山收橡实。</p>
<!—实现从左到右切换到下一页 -->
    <a href="#first" data-transition="slide" data-direction="reverse">上一页
</a>
  </div>
  <div data-role="footer">
    <h1>野老歌</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 17-13 所示。单击“下一页”超链接，即可从右到左滑动进入第二页，如图 17-14 所示。单击“上一页”超链接，即可从左到右滑动返回到第一页。



图 17-13　程序预览效果



图 17-14　第二页预览效果

17.6　高 手 解 惑

疑问 1：如何在模拟器中查看做好的网页效果？

答：HTML 文件制作完成后，要想在模拟器中测试，可以在地址栏中输入文件的路径，

例如输入如下：

```
file://localhost/ch16/16.2.html
```

为了防止输入错误，可以直接将文件拖曳到地址栏中，模拟器会自动帮助用户添加完整路径。

疑问 2：jQuery Mobile 支持哪些移动设备？

答：目前市面上移动设备非常多，如果想查询 jQuery Mobile 支持哪些移动设备，可以参照 jQuery Mobile 网站的各厂商支持表，还可以参考维基百科网站对 jQuery Mobile 说明中提供的 Mobile browser support 一览表。

疑问 3：我的浏览器为什么不支持页面切换效果？

答：为了实现页面切换效果，浏览器必须支持功能。目前，支持 CSS 3 3D 切换功能的浏览器最小版本包括 IE 10.0、Chrome 12.0、Firefox 16.0、Safari 4.0 和 Opera 15.0 等。



第 18 章

jQuery Mobile UI 组件

jQuery Mobile 针对用户界面提供了各种可视化的标签，包括按钮、复选框、选择菜单、列表、弹窗、工具栏、面板、导航和布局等。这些可视化标签与 HTML 5 标记一起使用，即可轻轻松松地开发出绚丽多彩的移动网页。本章重点学习这些标签的使用方法和技巧。



18.1　套用 UI 组件

jQuery Mobile 提供很多可视化的 UI 组件，只要套用之后，就可以生成绚丽且适合移动设备使用的组件。jQuery Mobile 中各种可视化的 UI 组件与 HTML 5 标记大同小异。下面介绍常用组件的用法，其中按钮、列表等功能变化比较大的后面会做详细介绍。

18.1.1　案例 1——表单组件

jQuery Mobile 使用 CSS 自动为 HTML 表单添加样式，让它们看起来更具吸引力，触摸起来更具友好性。

在 jQuery Mobile 中，经常使用的表单控件如下。

1. 文本框

文本框的语法格式如下：

```
<input type="text" name="fname" id="fname" value=" ">
```

其中 value 属性是文本框中显示的内容，也可以使用 placeholder 来指定一个简短的描述，用来描述输入内容的含义。

【例 18.1】 使用文本框(实例文件：ch18\18.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="first">
  <div data-role="header">
  <h1>输入会员信息</h1>
  </div>
  <div data-role="main" class="ui-content">
    <form>
      <div class="ui-field-contain">
        <label for="fullname">姓名：</label>
        <input type="text" name="fullname" id="fullname">
        <label for="bday">出生年月：</label>
        <input type="date" name="bday" id="bday">
        <label for="email">E-mail:</label>
        <input type="email" name="email" id="email" placeholder="输入您的电子
邮箱">
```

```
      </div>
      <input type="submit" data-inline="true" value="注册">
    </form>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-1 所示。单击“出生年月”下拉列表框时，会自动打开日期选择器，用户直接选择相应的日期即可，如图 18-2 所示。

图 18-1 程序预览效果



图 18-2 日期选择器

2. 文本域

使用<textarea>可以实现多行文本的输入。

【例 18.2】 使用文本域(实例文件：ch18\18.2.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="first">
  <div data-role="header">
  <h1>多行文本域</h1>
  </div>
  <div data-role="main" class="ui-content">
    <form>
       <div class="ui-field-contain">
```

```
        <label for="info">输入最喜欢的一首古诗:</label>
        <textarea name="addinfo" id="info"></textarea>
      </div>
      <input type="submit" data-inline="true" value="提交">
    </form>
     </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-3 所示。输入多行内容时，文本域会根据输入的内容，自动调整高度，如图 18-4 所示。



图 18-3　程序预览效果



图 18-4　输入内容

3. 搜索输入框

HTML 5 中新增的 type="search"类型为搜索输入框，它是为输入搜索定义文本字段。

搜索输入框的语法格式如下：

```
<input type="search" name="search" id="search" placeholder="搜索内容">
```

搜索输入框的效果如图 18-5 所示。

图 18-5　搜索输入框

4. 范围滑动条

使用<input type="range">控件，即可创建范围滑动条，语法格式如下：

```
<input type="range" name="points" id="points" value="50" min="0" max="100"
data-show-value="true">
```



其中，max 属性规定允许的最大值。min 属性规定允许的最小值。step 属性规定合法的数字间隔。value 属性规定默认值。data-show-value 属性规定是否在按钮上显示进度的值，如果设置为 true，则表示显示进度的值；如果设置为 false，则表示不显示进度的值。

【例 18.3】 使用范围滑动条(实例文件：ch18\18.3.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="first">
  <div data-role="header">
    <h1>工作进度申报</h1>
  </div>
  <div data-role="main" class="ui-content">
    <form>
      <label for="points">工作完成的进度:</label>
      <input type="range" name="points" id="points"
value="50" min="0" max="100" data-show-
value="true">
      <input type="submit" data-inline="true"
value="提交">
    </form>
  </div>
</div>
</body>
</html>
```

图 18-6　程序预览效果

在模拟器中的预览效果如图 18-6 所示。用户可以拖动滑块，选择需要的值。也可以通过加减按钮，精确选择进度的值。

使用 data-popup-enabled 属性可以设置小弹窗效果，代码如下：

```
<input type="range" name="points" id="points" value="50" min="0" max="100"
data-popup-enabled="true">
```

添加后的效果如图 18-7 所示。



图 18-7　进度值显示效果

使用 data-highlight 属性可以高亮度显示滑动条的值，代码如下：

```
<input type="range" name="points" id="points" value="50" min="0" max="100"
data-highlight="true">
```

添加后的效果如图 18-8 所示。



图 18-8　高亮度显示进度值效果

5. 表单按钮

表单按钮分为 3 种，即提交按钮、取消按钮和普通按钮。只需要在 type 属性中设置表单的类型即可，代码如下：

```
<input type="submit" value="提交按钮">
<input type="reset" value="取消按钮">
<input type="button" value="普通按钮">
```

在模拟器中的预览效果如图 18-9 所示。

当用户在有限数量的选择中仅选取一个选项时，经常用到表单中的单选按钮。通过 type="radio"来创建一系列单选按钮，代码如下：

```
<fieldset data-role="controlgroup">
<legend>请选择您的年级：</legend>
   <label for="one">一年级</label>
   <input type="radio" name="grade" id="one" value="one">
   <label for="two">二年级</label>
   <input type="radio" name="grade" id="two" value="two">
   <label for="three">三年级</label>
   <input type="radio" name="grade" id=" three" value=" three">
</fieldset>
```

在模拟器中的预览效果如图 18-10 所示。



图 18-9　表单按钮预览效果



图 18-10　单选按钮



<fieldset>标记用来创建按钮组，组内各个组件保持自己的功能。在<fieldset>标记内添加 data-role="controlgroup"，这样这些单选按钮样式统一，看起来像一个组合。其中<legend>标签用来定义按钮组的标题。



6. 复选框

当用户在有限数量的选择中选取一个或多个选项时，需要使用复选框。代码如下：

```
<fieldset data-role="controlgroup">
  <legend>请选择您喜爱的季节：</legend>
  <label for="spring">春天</label>
  <input type="checkbox" name="season" id="spring" value="spring">
  <label for="summer">夏天</label>
  <input type="checkbox" name="season" id="summer" value="summer">
  <label for="fall">秋天</label>
  <input type="checkbox" name="season" id="fall" value="fall">
  <label for="winter">冬天</label>
  <input type="checkbox" name="season" id="winter" value="winter">
 </fieldset>
```

在模拟器中的预览效果如图 18-11 所示。



图 18-11 复选框

7. 选择菜单

使用<select>标签可以创建带有若干选项的下拉列表。<select>标签内的<option>属性定义了列表中的可用选项。代码如下：

```
<fieldset data-role="fieldcontain">
      <label for="day">选择值日时间：</label>
      <select name="day" id="day">
       <option value="mon">星期一</option>
       <option value="tue">星期二</option>
       <option value="wed">星期三</option>
       <option value="thu">星期四</option>
       <option value="fri">星期五</option>
       <option value="sat">星期六</option>
       <option value="sun">星期日</option>
      </select>
</fieldset>
```



图 18-12 选择菜单

在模拟器中的预览效果如图 18-12 所示。

如果菜单中的选项还需要再次分组，可以在<select>内使用<optgroup>标签，添加后的代码如下：

```
<fieldset data-role="fieldcontain">
      <label for="day">选择值日时间：</label>
```

```
        <select name="day" id="day">
        <optgroup label="工作日">
         <option value="mon">星期一</option>
         <option value="tue">星期二</option>
         <option value="wed">星期三</option>
         <option value="thu">星期四</option>
         <option value="fri">星期五</option>
        </optgroup>
        <optgroup label="休息日">
         <option value="sat">星期六</option>
         <option value="sun">星期日</option>
        </optgroup>
        </select>
</fieldset>
```



图 18-13　菜单选项分组后的效果

在模拟器中的预览效果如图 18-13 所示。

如果想选择菜单中的多个选项，需要设置<select>标签的 multiple 属性。设置代码如下：

```
<select name="day" id="day" multiple data-native-menu="false">
```

例如把上面的代码修改如下：

```
<fieldset data-role="fieldcontain">
        <label for="day">选择多个值日时间：</label>
        <select name="day" id="day" multiple data-native-menu="false">
        <optgroup label="工作日">
         <option value="mon">星期一</option>
         <option value="tue">星期二</option>
         <option value="wed">星期三</option>
         <option value="thu">星期四</option>
         <option value="fri">星期五</option>
        </optgroup>
        <optgroup label="休息日">
         <option value="sat">星期六</option>
         <option value="sun">星期日</option>
        </optgroup>
        </select>
</fieldset>
```

在模拟器中预览，选择菜单时的效果如图 18-14 所示。选择完成后，即可看到多个菜单选项被选择，如图 18-15 所示。



图 18-14　选中多个菜单选项

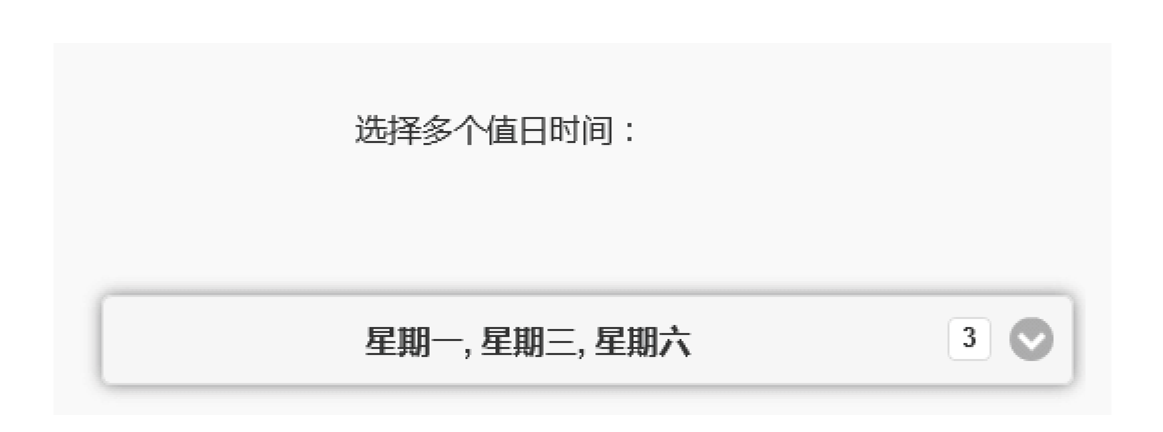


图 18-15　多个菜单选项被选择后的效果



8. 翻转波动开关

设置<input type="checkbox">标签的 data-role 为"flipswitch"时，可以创建翻转波动开关。代码如下：

```
<form>
   <label for="switch">切换开关：</label>
    <input type="checkbox" data-role="flipswitch" name="switch"
id="switch">
</form>
```

在模拟器中的预览效果如图 18-16 所示。

同时，用户还可以使用"checked"属性来设置默认的选项。代码如下：

```
<input type="checkbox" data-role="flipswitch" name="switch" id="switch"
checked>
```

修改后的预览效果如图 18-17 所示。

在默认情况下，开关切换的文本为 On 和 Off。可以使用 data-on-text 和 data-off-text 属性来修改，代码如下：

```
<input type="checkbox" data-role="flipswitch" name="switch" id="switch"
data-on-text="打开" data-off-text="关闭">
```

修改后的预览效果如图 18-18 所示。



图 18-16　开关默认效果　　图 18-17　修改默认选项后的效果　　图 18-18　修改切换开关文本后的效果

18.1.2　案例 2——按钮和组按钮

前面简单介绍过表单按钮。由于按钮和按钮组功能变化比较大，下面详细讲述它们的使用方法和技巧。

在 jQuery Mobile 中，创建按钮的方法包括以下 3 种。

(1)　使用<button>标签创建普通按钮。代码如下：

```
<button>按钮</button>
```

(2)　使用<input>标签创建表单按钮。代码如下：

```
<input type="button" value="按钮">
```

(3)　使用 data-role="button"属性创建链接按钮。代码如下：

```
<a href="#" data-role="button">按钮</a>
```

在 jQuery Mobile 中，按钮的样式会被自动添加上。为了让按钮在移动设备上更具吸引力和可用性，推荐在页面间进行链接时，使用第三种方法；在表单提交时，使用第一种或第二种方法。

在默认情况下，按钮占满整个屏幕宽度。如果想要一个仅是与内容一样宽的按钮，或者需要并排显示两个或多个按钮，可以通过设置 data-inline="true"来完成。代码如下：

```
<a href="#pagetwo" data-role="button" data-inline="true">下一页</a>
```

下面通过一个案例来区别默认按钮和设置后按钮的区别。

【例 18.4】 使用不同的按钮(实例文件：ch18\18.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>按钮的区别</h1>
  </div>
  <div data-role="content" class="content">
    <p>普通 / 默认按钮:</p>
    <a href="#second" data-role="button">下一页</a>
    <p>设置后的按钮:</p>
    <a href="#second" data-inline="true">下一页</a>
    <a href="#first" data-inline="true">上一页</a>
  </div>
  <div data-role="footer">
    <h1>2 种按钮</h1>
  </div>
</div>
</body>
</html>
```

按钮的区别
普通 / 默认按钮:
下一页
设置后的按钮:
下一页 上一页
2种按钮

图 18-19　不同的按钮的效果

在模拟器中的预览效果如图 18-19 所示。

jQuery Mobile 提供了一个简单的方法来将按钮组合在一起。使用 data-role="controlgroup"属性即可通过按钮组来组合按钮。同时使用 data-type="horizontal|vertical"属性来设置按钮的排列方式是水平还是垂直。

【例 18.5】 组按钮的排列(实例文件：ch18\18.5.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
```

```
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>组按钮的排列</h1>
  </div>
  <div data-role="content" class="content">
<div data-role="controlgroup" data-type="horizontal">
    <p>水平排列的按钮：</p>
    <a href="#" data-role="button">按钮 a</a>
    <a href="#" data-role="button">按钮 b</a>
    <a href="#" data-role="button">按钮 c</a>
</div><br>
    <div data-role="controlgroup" data-type="vertical"
    <p>垂直排列的按钮:</p>
     <a href="#" data-role="button">按钮 a</a>
    <a href="#" data-role="button">按钮 b </a>
    <a href="#" data-role="button">按钮 c</a>
</div>
  </div>
  <div data-role="footer">
    <h1>2 种排列方式</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-20 所示。

图 18-20　不同排列方式的按钮组

18.1.3　案例 3——按钮图标

jQuery Mobile 提供了一套丰富多彩的按钮图标。用户只需要使用 data-icon 属性即可添加按钮图标。常用的图标样式如表 18-1 所示。

表 18-1　常用的按钮图标样式

图标参数	外观样式	说　明
data-icon="arrow-l"	左箭头	左箭头
data-icon="arrow-r"	右箭头	右箭头
data-icon="arrow-u"	上箭头	上箭头
data-icon="arrow-d"	下箭头	下箭头
data-icon="info"	信息	信息
data-icon="plus"	加号	加号
data-icon="minus"	减号	减号
data-icon="check"	复选	复选
data-icon="refresh"	重新整理	重新整理
data-icon="delete"	删除	删除
data-icon="forward"	前进	前进
data-icon="back"	后退	后退
data-icon="star"	星形	星形
data-icon="audio"	扬声器	扬声器
data-icon="lock"	挂锁	挂锁
data-icon="search"	搜索	搜索
data-icon="alert"	警告	警告
data-icon="grid"	网格	网格
data-icon="home"	首页	首页

例如以下代码：

```
<a href="#" data-role="button" data-icon="lock">挂锁</a>
<a href="#" data-role="button" data-icon="check">复选</a>
<a href="#" data-role="button" data-icon="refresh">重新整理</a>
<a href="#" data-role="button" data-icon="delete">删除</a>
```

在模拟器中的预览效果如图 18-21 所示。

细心的读者会发现，按钮上的图标默认情况下会出现在按钮的左边。如果需要设置图标的位置，可以设置 data-iconpos 属性来指定位置，包括 top(顶部)、right(右侧)和 bottom(底部)。例如以下代码：

```
<a href="#" data-role="button" data-icon="refresh">重新整理</a>
<a href="#" data-role="button" data-icon="refresh" data-iconpos="top">重新整
理</a>
<a href="#" data-role="button" data-icon="refresh" data-iconpos="right">重
新整理</a>
<a href="#" data-role="button" data-icon="refresh" data-iconpos="bottom">重
新整理</a>
```

在模拟器中的预览效果如图 18-22 所示。



图 18-21　按钮图标效果



图 18-22　设置图标的位置



如果不想让按钮上出现文字，可以将 data-iconpos 属性设置为 notext，这样只会显示按钮，而没有文字。

18.1.4　案例 4——弹窗

弹窗是一个非常流行的对话框。弹窗可以覆盖在页面上展示。弹窗可用于显示一段文本、图片、地图或其他内容。创建一个弹窗，需要使用<a>和<div>标签。在<a>标签中添加 data-rel="popup"属性，在<div>标签中添加 data-role="popup"属性。然后为<div>设置 id，设置<a>的 href 值为<div>指定的 id，其中<div>中的内容为弹窗显示的内容。代码如下：

```
<a href="#firstpp" data-rel="popup">显示弹窗</a>
<div data-role="popup" id="firstpp">
  <p>这是弹出窗口显示的内容</p>
</div>
```

在模拟器中的预览效果如图 18-23 所示。单击“显示弹窗”按钮即可显示弹出窗口的内容。



<div>弹窗与点击的<a>链接必须在同一个页面上。

在默认情况下，点击弹窗之外的区域或按 Esc 键即可关闭弹窗。用户也可以在弹窗上添加关闭按钮，只需要设置属性 data-rel="back"即可，结果如图 18-24 所示。



图 18-23　弹窗效果



图 18-24　带关闭按钮的弹窗效果

用户还可以在弹窗中显示图片，代码如下：

```
<div id="pageone" data-role="content" class="content" >
    <p>单击下面的小图片</p>
    <a href="#firstpp" data-rel="popup" >
    <img src="123.jpeg" style="width:200px;"></a>
    <div data-role="popup" id="firstpp">
```

```
    <p>这是我的图片！</p>
    </a><img src="123.jpeg" style="width:500px;height:500px;" >
    </div>
  </div>
```

在模拟器中的预览效果如图 18-25 所示。单击图片，即可弹出如图 18-26 所示的图片弹窗。

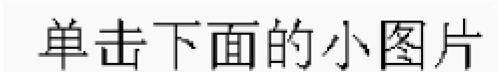




图 18-25　在模拟器中的预览效果



图 18-26　图片弹窗效果

18.2　列　　表

与电脑相比，移动设备屏幕比较小，所以常常以列表的形式显示数据。下面学习列表的使用方法和技巧。

18.2.1　案例 5——列表视图

jQuery Mobile 中的列表视图是标准的 HTML 列表，包括有序列表<ol>和无序列表<ul>。列表视图是 jQuery Mobile 中功能强大的一个特性。它会使标准的无序或有序列表应用更广泛。

列表的使用方法非常简单，只需要在<ul>或<ol>标签中添加属性 data-role="listview"。每个项目(<li>)中可以添加链接。下面通过一个案例来学习。

【例 18.6】 使用列表视图(实例文件：ch18\18.6.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>列表视图</h1>
```

```
  </div>
  <div data-role="content" class="content">
  <h2>有序列表：</h2>
    <ol data-role="listview">
      <li><a href="#">香蕉</a></li>
      <li><a href="#">橘子</a></li>
      <li><a href="#">苹果</a></li>
    </ol>
    <h2>无序列表：</h2>
    <ul data-role="listview">
      <li><a href="#">芹菜</a></li>
      <li><a href="#">韭菜</a></li>
      <li><a href="#">菠菜</a></li>
    </ul>
</div>
  </div>
  <div data-role="footer">
    <h1>有序列表和无序列表</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-27 所示。



图 18-27　有序列表和无序列表



在默认情况下，列表项的链接会自动变成一个按钮，此时不再需要使用 data-role="button"属性。

从结果可以看出，列表样式中没有边缘和圆角效果，这里可以通过设置属性 data-inset="true"来完成，代码如下：

```
<ul data-role="listview" data-inset="true">
```

上面案例的部分代码修改如下：

```
  <div data-role="content" class="content">
  <h2>标准列表样式：</h2>
    <ol data-role="listview">
      <li><a href="#">香蕉</a></li>
```

```
    <li><a href="#">橘子</a></li>
    <li><a href="#">苹果</a></li>
  </ol>
  <h2>添加 data-inset="true"属性后的样式：</h2>
  <ul data-role="listview" data-inset="true">
    <li><a href="#">芹菜</a></li>
    <li><a href="#">韭菜</a></li>
    <li><a href="#">菠菜</a></li>
  </ul>
</div>
```

在模拟器中的预览效果如图 18-28 所示。

如果列表项比较多，用户可以使用列表分隔项对列表进行分组操作，这样使列表看起来更整齐。通过在列表项<li>标签中添加 data-role="list-divider" 属性即可指定列表分隔，例如以下代码：

```
<ul data-role="listview">
 <li data-role="list-divider">蔬菜</li>
  <li><a href="#">芹菜</a></li>
  <li><a href="#">韭菜</a></li>
<li data-role="list-divider">水果</li>
  <li><a href="#">苹果</a></li>
  <li><a href="#">橘子</a></li>
<li data-role="list-divider">乳制品</li>
  <li><a href="#">酸奶</a></li>
  <li><a href="#">奶酪</a></li>
 </ul>
```

在模拟器中的预览效果如图 18-29 所示。



图 18-28　有边缘和圆角的列表效果



图 18-29　对项目进行分隔后的效果

如果项目列表是一个按字母顺序排列的列表，通过添加 data-autodividers="true"属性，可以自动生成项目的分隔。代码如下：

```
<ul data-role="listview" data-autodividers="true">
  <li><a href="#">Avocado</a></li>
  <li><a href="#"> Apricot</a></li>
  <li><a href="#">Banana</a></li>
  <li><a href="#">Bramley</a></li>
  <li><a href="#"> Cherry </a></li>
</ul>
```

在模拟器中的预览效果如图 18-30 所示。从结果可以看出，创建的分隔文本是列表项文本的第一个大写字母。



图 18-30　自动生成分隔后的效果

18.2.2　案例 6——列表内容

在列表内容中，既可以添加图片和说明，也可以添加计数泡泡，同时还能拆分按钮和列表的链接。

1. 加入图片和说明

在前面做的案例中，列表项目前没有图片或说明。下面讲述如何添加图片和说明。代码如下：

```
<li>
   <a href="#">
   <img src="124.jpg">
   <h3>香蕉</h3>
   <p>香蕉的原产地是东南亚</p>
   </a>
</li>
```

在模拟器中的预览效果如图 18-31 所示。

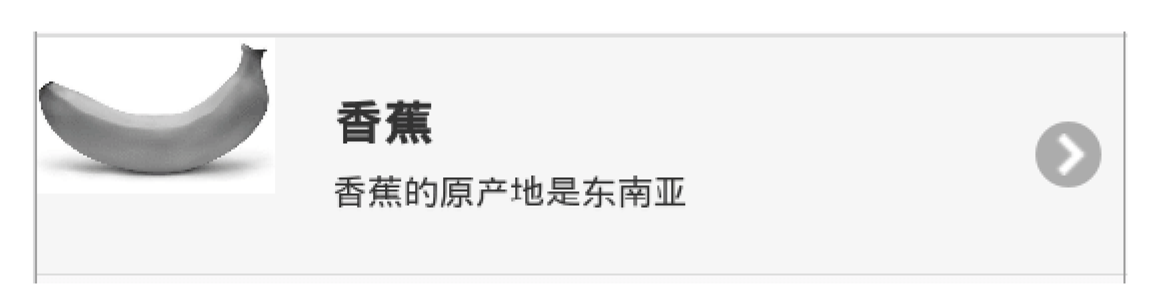


图 18-31　加入图片和说明

2. 加入计数泡泡

计数泡泡主要是在列表中显示数字时使用，只需要在<li>标签中加入以下标签即可：

```
<span class="ui-li-count">数字</span>
```

例如：

```
<li>
   <a href="#">
   <img src="124.jpg">
   <h3>香蕉</h3>
```

```
    <p>香蕉的原产地是东南亚</p>
    <span class="ui-li-count">111</span>
    </a>
</li>
```

在模拟器中的预览效果如图 18-32 所示。

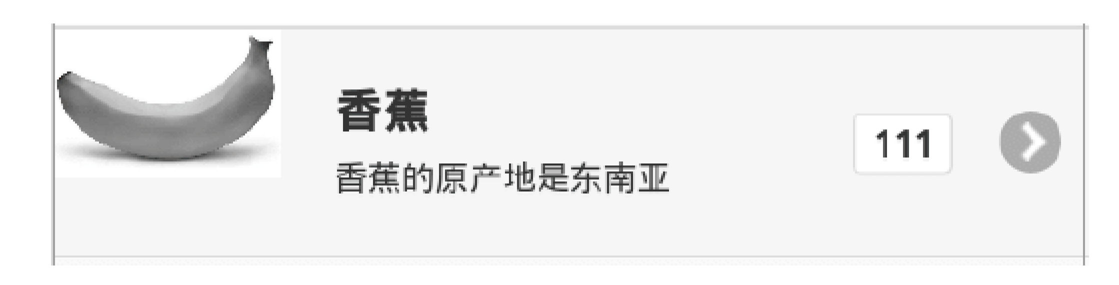


图 18-32　加入计数泡泡

3. 拆分按钮和列表的链接

在默认情况下，单击列表项或按钮，都是转向同一个链接。用户也可以拆分按钮和列表项的链接，这样单击按钮和列表项时，会转向不同的链接。设置方法比较简单，只需要在<li>标签中加入两组<a>标签即可。

例如：

```
<li>
    <a href="122.html">
    <img src="124.jpg">
    <h3>香蕉</h3>
        <p>香蕉的原产地是东南亚</p>
    </a>
    <a href="123.html data-icon="star"></a>
</li>
```

在模拟器中的预览效果如图 18-33 所示。

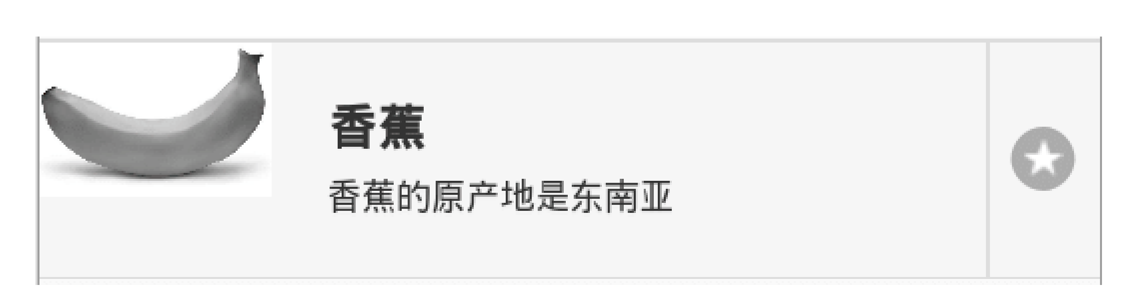


图 18-33　拆分按钮和列表的链接

18.2.3　案例 7——列表过滤

在 jQuery Mobile 中，用户可以对列表项目进行搜索过滤。添加过滤效果的具体步骤如下。

(1) 创建一个表单，并添加类 ui-filterable，该类的作用是自动调整搜索字段与过滤元素的外边距。代码如下：

```
<form class="ui-filterable">
</form>
```

(2) 在<form>标签内创建一个<input>标签，并添加 data-type="search"属性，并指定 id，从而创建基本的搜索字段。代码如下：

```
<form class="ui-filterable">
```



```
  <input id="myFilter" data-type="search">
</form>
```

(3) 为过滤的列表添加 data-input 属性，该值为<input>标签的 id。代码如下：

```
<ul data-role="listview" data-filter="true" data-input="#myFilter">
```

下面通过一个案例来理解列表是如何过滤的。

【例 18.7】 使用列表过滤(实例文件：ch18\18.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="content" class="content">
   <h2>进货商联系表</h2>
   <form>
    <input id="myFilter" data-type="search">
   </form>
    <ul data-role="listview" data-filter="true" data-input="#myFilter">
      <li><a href="#">张小名</a></li>
      <li><a href="#">刘名园</a></li>
      <li><a href="#">刘鲲鹏</a></li>
     <li><a href="#">张鹏举</a></li>
     <li><a href="#">张鹏远</a></li>
    </ul>
   </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-34 所示。输入需要过滤的关键字，例如这里搜索姓张的进货商，结果如图 18-35 所示。

进货商联系表

张小名

刘名园

刘鲲鹏

张鹏举

张鹏远

图 18-34 程序预览效果



图 18-35 列表过滤后的效果



如果需要在搜索框内添加提示信息，可以通过设置 placeholder 属性来完成。代码如下：

```
    <input id="myFilter" data-type="search" placeholder="请输入联系人的姓">
```

18.3 面板和可折叠块

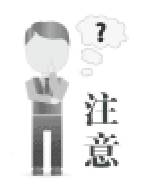
在 jQuery Mobile 中，可以通过面板或可折叠块来隐藏或显示指定的内容。下面重点学习面板和可折叠块的使用方法和技巧。

18.3.1 案例 8——面板

在 jQuery Mobile 中可以添加面板，面板会在屏幕上从左到右划出。通过为<div>标签添加 data-role="panel"属性来创建面板。具体方法如下。

(1) 通过<div>标签来定义面板的内容，并定义 id 属性。例如以下代码：

```
<div data-role="panel" id="myPanel">
    <h2>长恨歌</h2>
    <p>天生丽质难自弃，一朝选在君王侧。回眸一笑百媚生，六宫粉黛无颜色。</p>
</div>
```



定义的面板内容必须置于头部、内容和底部组成的页面之前或之后。

(2) 要访问面板，需要创建一个指向面板<div>的链接，单击该链接即可打开面板。例如以下代码：

```
<a href="#myPanel" class="ui-btn ui-btn-inline">最喜欢的诗句</a>
```

【例 18.8】创建指向面板的链接(实例文件：ch18\18.8.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="first">
  <div data-role="panel" id="myPanel">
    <h2>长恨歌</h2>
    <p>天生丽质难自弃，一朝选在君王侧。回眸一笑百媚生，六宫粉黛无颜色。</p>
  </div>
  <div data-role="header">
```

```
  <h1>使用面板</h1>
  </div>
  <div data-role="content" class="content">
    <a href="#myPanel" class="ui-btn ui-btn-inline">最喜欢的诗句</a>
   </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-36 所示。单击“最喜欢的诗句”超链接，即可打开面板，结果如图 18-37 所示。



图 18-36　程序预览效果

长恨歌

天生丽质难自弃，一朝选在君王侧。回眸一笑百媚生，六宫粉黛无颜色。

图 18-37　打开面板

面板的展示方式由属性 data-display 来控制，分为以下 3 种。

(1) data-display="reveal"：面板的展示方式为从左到右划出，这是面板展示方式的默认值。

(2) data-display="overlay"：在内容上显示面板。

(3) data-display="push"：同时“推动”面板和页面。

这 3 种面板展示方式的代码如下：

```
<div data-role="panel" id="overlayPanel" data-display="overlay">
<div data-role="panel" id="revealPanel" data-display="reveal">
<div data-role="panel" id="pushPanel" data-display="push">
```

在默认情况下，面板会显示在屏幕的左侧。如果想让面板出现在屏幕的右侧，可以指定 data-position="right" 属性。代码如下：

```
<div data-role="panel" id="myPanel" data-position="right">
```

在默认情况下，面板是随着页面一起滚动的。如果你需要实现面板内容固定、不随页面滚动而滚动，可以为面板添加 the data-position-fixed="true"属性。代码如下：

```
<div data-role="panel" id="myPanel" data-position-fixed="true">
```

18.3.2　案例 9——可折叠块

通过可折叠块，用户可以隐藏或显示指定的内容，这对于存储部分信息很有用。

创建可折叠块的方法比较简单，只需要在<div>标签中添加 data-role="collapsible" 属性即

可，添加标题标签为 H1-H6，后面即可添加隐藏的信息。

```
<div data-role="collapsible">
 <h1>折叠块的标题</h1>
 <p>可折叠的具体内容。</p>
 </div>
```

【例 18.9】 创建可折叠块(实例文件：ch18\18.9.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="first">
  <div data-role="header">
  <h1>可折叠块</h1>
  </div>
  <div data-role="content" class="content">
      <div data-role="collapsible">
         <h1>最喜欢的水果</h1>
         <p>香蕉、橘子、苹果</p>
         </div>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-38 所示。单击“最喜欢的水果”按钮，即可打开可折叠块，结果如图 18-39 所示。



图 18-38 程序预览效果



图 18-39 打开可折叠块



在默认情况下，内容是被折叠起来的。如果需要在页面加载时展开内容，添加 data-collapsed="false"属性即可。代码如下：

```
<div data-role="collapsible" data-collapsed="false">
 <h1>折叠块的标题</h1>
 <p>这里显示的内容是展开的</p>
 </div>
```



可折叠块是可以嵌套的，例如以下代码：

```
<div data-role="collapsible">
 <h1>全部智能商品</h1>
<div data-role="collapsible">
 <h1>智能家居</h1>
 <p>智能办公、智能厨电和智能网络</p>
 </div>
</div>
```

在模拟器中的预览效果如图 18-40 所示。



图 18-40　程序预览效果

18.4　案例 10——导航条

导航条通常位于页面的头部或尾部，主要作用是便于用户快速访问需要的页面。下面重点学习导航条的使用方法和技巧。

在 jQuery Mobile 中，使用 data-role="navbar" 属性来定义导航栏。需要特别注意的是，导航栏中的链接将自动变成按钮，不需要使用 data-role="button"属性。

例如以下代码：

```
<div data-role="header">
  <h1>鸿鹄网购平台</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#">主页</a></li>
      <li><a href="#">团购</a></li>
      <li><a href="#">搜索商品</a></li>
    </ul>
  </div>
</div>
```

在模拟器中的预览效果如图 18-41 所示。



图 18-41　程序预览效果

通过前面章节的学习，用户还可以为导航添加按钮图标，例如以上代码修改如下：

```
<div data-role="header">
  <h1>鸿鹄网购平台</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#" data-icon="home">主页</a></li>
      <li><a href="#" data-icon="arrow-d">团购</a></li>
      <li><a href="#" data-icon="search">搜索商品</a></li>
    </ul>
  </div>
</div>
```

在模拟器中的预览效果如图 18-42 所示。



图 18-42　程序预览效果

细心的读者会发现，导航按钮的图标默认位置是位于文字的上方，这个普通的按钮图片是不一样的。如果需要修改导航按钮图标的位置，可以通过设置 data-iconpos 属性来指定位置，包括 left(左侧)、right(右侧)和 bottom(底部)。

例如，下面修改导航按钮图标的位置为文本的左侧，代码如下：

```
<div data-role="header">
  <h1>鸿鹄网购平台</h1>
  <div data-role="navbar" data-iconpos="left">
    <ul>
      <li><a href="#" data-icon="home" >主页</a></li>
      <li><a href="#" data-icon="arrow-d" >团购</a></li>
      <li><a href="#" data-icon="search">搜索商品</a></li>
    </ul>
  </div>
</div>
```

在模拟器中的预览效果如图 18-43 所示。



图 18-43　程序预览效果



和设置普通按钮图标位置不同的是，这里 data-iconpos="left"属性只能添加到<div>标签中，而不能添加到<li>标签中，否则是无效的，读者可以自行检测。

在默认情况下，当单击导航按钮时，按钮的样式会发生变化。例如这里单击“搜索商品”导航按钮，发现按钮的底纹颜色变成了蓝色，如图 18-44 所示。

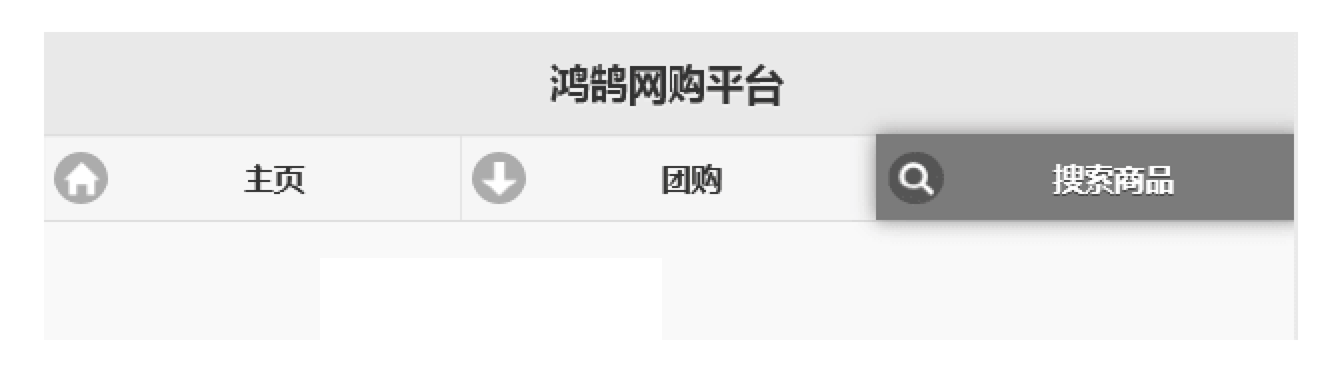


图 18-44　导航按钮的样式变化

如果用户想取消上面的样式变化，可以添加 class="ui-btn-active"属性即可。例如以下代码：



```
<li><a href="#anylink" class="ui-btn-active">首页</a></li>
```

修改完成后，再次单击“首页”导航按钮时，样式不会发生变化。

对于多个页面的情况，往往用户希望显示哪个页面，对应导航按钮处于选中状态。下面通过一个案例来讲解。

【例 18.10】 对应于显示的页面，导航按钮处于选中状态(实例文件：ch18\18.10.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
  <h1>鸿鹄购物平台</h1>
    <div data-role="navbar">
      <ul>
        <li><a href="#" class="ui-btn-active ui-state-persist">主页</a></li>
        <li><a href="#second">团购</a></li>
        <li><a href="#">搜索商品</a></li>
      </ul>
    </div>
  </div>
  <div data-role="content" class="content">
<p>这里是首页显示的内容</p>
 </div>
 <div data-role="footer">
    <h1>首页</h1>
  </div>
</div>

<div data-role="page" id="second">
  <div data-role="header">
  <h1>鸿鹄购物平台</h1>
    <div data-role="navbar">
      <ul>
        <li><a href="#first">主页</a></li>
        <li><a href="#" class="ui-btn-active ui-state-persist">团购</a></li>
        <li><a href="#">搜索商品</a></li>
      </ul>
  </div>
  </div>
  <div data-role="content" class="content">
<p>这里是团购显示的内容</p>
 </div>
 <div data-role="footer">
    <h1>团购页面</h1>
  </div>
```

```
</div>
</body>
</html>
```

在模拟器中的预览效果如图 18-45 所示。此时默认显示首页的内容，“主页”导航按钮处于选中状态。切换到团购页面后，“团购”导航按钮处于选中状态，如图 18-46 所示。



图 18-45　程序预览效果



图 18-46　“团购”导航按钮处于选中状态

18.5　综合案例——使用 jQuery Mobile 主题

用户在设计移动网站时，往往需要配置背景颜色、导航颜色、布局颜色等，这些工作是非常耗费时间的。为此，jQuery Mobile 提供两种不同的主题样式，每种主题颜色的按钮、导航、内容等颜色都是配置好的，效果也不相同。

这两种主题分别为 a 和 b，通过设置 data-theme 属性来引用主题 a 或 b。代码如下：

```
<div data-role="page" id="first" data-theme="a">
<div data-role="page" id="first" data-theme="b">
```

1. 主题 a

页面为灰色背景、黑色文字；头部与底部均为灰色背景、黑色文字；按钮为灰色背景、黑色文字；激活的按钮和链接为白色文本、蓝色背景；input 输入框中 placeholder 属性值为浅灰色，value 值为黑色。

下面通过一个案例来讲解主题 a 的样式效果。

【例 18.11】 主题 a 的样式(实例文件：ch18\18.11.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
 </head>
<body>
<div data-role="page" id="first" data-theme="a">
```

```
  <div data-role="header">
    <h1>古诗鉴赏</h1>
  </div>

  <div data-role="content " class="content">
    <p>秋风起兮白云飞，草木黄落兮雁南归。兰有秀兮菊有芳，怀佳人兮不能忘。泛楼船兮济汾河，
     横中流兮扬素波。</p>
    <a href="#">秋风辞</a>
    <a href="#" class="ui-btn">更多古诗</a>
    <p>唐诗:</p>
    <ul data-role="listview" data-autodividers="true" data-inset="true">
      <li><a href="#">将进酒</a></li>
      <li><a href="#">春望</a></li>
    </ul>
    <label for="fullname">请输入喜欢诗的名字:</label>
      <input type="text" name="fullname" id="fullname" placeholder="诗词名
称..">
    <label for="switch">切换开关:</label>
      <select name="switch" id="switch" data-role="slider">
        <option value="on">On</option>
        <option value="off" selected>Off</option>
      </select>
  </div>

  <div data-role="footer">
    <h1>经典诗歌</h1>
  </div>
</div>
</body>
</html>
```

主题 a 的样式效果如图 18-47 所示。

图 18-47　主题 a 样式效果

2. 主题 b

页面为黑色背景、白色文字；头部与底部均为黑色背景、白色文字；按钮为白色文字、

木炭背景；激活的按钮和链接为白色文本、蓝色背景；input 输入框中 placeholder 属性值为浅灰色、value 值为白色。

为了对比主题 a 的样式效果，请将上面案例中的代码：

```
<div data-role="page" id="first" data-theme="a">
```

修改如下：

```
<div data-role="page" id="first" data-theme="b">
```

主题 b 的样式效果如图 18-48 所示。

图 18-48　主题 b 样式效果

主题样式 a 和 b 不仅仅可以应用到页面，也可以单独地应用到页面的头部、内容、底部、导航条、按钮、面板、列表、表单等元素上。

例如，将主题样式 b 添加到页面的头部和底部。代码如下：

```
<div data-role="header" data-theme="b"></div>
<div data-role="footer" data-theme="b"></div>
```

将主题样式 b 添加到对话框的头部和底部。代码如下：

```
<div data-role="page" data-dialog="true" id="second">
  <div data-role="header" data-theme="b"></div>
  <div data-role="footer" data-theme="b"></div>
</div>
```

将主题样式 b 添加到按钮上时，需要使用 class="ui-btn- a|b "来设置按钮颜色为灰色或黑

色。例如，将主题样式 b 应用到按钮上，代码如下：

```
<a href="#" class="ui-btn">灰色按钮(默认)</a>
<a href="#" class="ui-btn ui-btn-b">黑色按钮</a>
```

预览效果如图 18-49 所示。



图 18-49 按钮添加主题样式后的效果

在弹窗上应用主题样式的代码如下：

```
<div data-role="popup" id="myPopup" data-theme="b">
```

在头部和底部的按钮上也可以添加主题样式，例如以下代码：

```
<div data-role="header">
  <a href="#" class="ui-btn ui-btn-b">主页</a>
  <h1>古诗欣赏</h1>
  <a href="#" class="ui-btn">搜索</a>
</div>

<div data-role="footer">
   <a href="#" class="ui-btn ui-btn-b">上传古诗图文</a>
  <a href="#" class="ui-btn">名句欣赏鉴别</a>
  <a href="#" class="ui-btn ui-btn-b">联系我们</a>
</div>
```

预览效果如图 18-50 所示。



图 18-50 头部和底部的按钮添加主题样式后的效果

18.6 高 手 解 惑

疑问 1：如何制作一个后退按钮？

答： 如果需要创建后退按钮，请使用 data-rel="back" 属性(这会忽略锚的 href 值)：

```
<a href="#" data-role="button" data-rel="back">返回</a>
```

疑问 2：如何在面板上添加主题样式 b？

答：在面板上添加主题样式的方法比较简单。代码如下：

```
<div data-role="panel" id="myPanel" data-theme="b">
```

面板添加主题样式 b 后的效果如图 18-51 所示。



图 18-51　面板添加主题样式后的效果



第 19 章 jQuery Mobile 事件

页面有了事件就有了“灵魂”，可见事件对于页面是多么重要。这是因为事件使页面具有动态性和响应性，如果没有事件将很难完成页面与用户之间的交互。jQuery Mobile 针对移动端提供了各种浏览器事件，包括页面事件、触摸事件、滑动事件、定位事件等。本章介绍如何使用 jQuery Mobile 的事件。



19.1 页 面 事 件

jQuery Mobile 针对各个页面生命周期的事件可以分为以下几种。

(1) 初始化事件。分别在页面初始化之前，页面创建时和页面初始化之后触发的事件。

(2) 外部页面加载事件。外部页面加载时触发事件。

(3) 页面过渡事件。页面过渡时触发事件。

使用 jQuery Mobile 事件的方法比较简单，只需要使用 on()方法指定要触发的时间并设定事件处理函数即可，语法格式如下：

```
$(document).on(事件名称,选择器,事件处理函数)
```

其中选择器可选参数，如果省略该参数，表示事件应用于整个页面而不限定哪一个组件。

19.1.1 案例 1——初始化事件

初始化事件发生的时间包括页面初始化之前、页面创建时和页面创建后。下面详细介绍初始化事件。

1. Mobileinit

当 jQuery Mobile 开始执行时，首先会触发 mobileinit 事件。如果想更改 jQuery Mobile 的默认值，就可以将函数绑定到 mobileinit 事件。其语法格式如下：

```
$(document).on("mobileinit",function(){
   // jQuery 事件
});
```

例如，jQuery Mobile 开始执行任何操作时都会使用 Ajax 的方式。如果不想使用 Ajax，可以在 mobileinit 事件中将$.mobile.ajaxEnabled 更改为 false，代码如下：

```
$(document).on("mobileinit",function(){
  $.mobile.ajaxEnabled=false;
});
```

这里需要注意的是，上面的代码要放在引用 jquery.mobile.js 之前。

2. jQuery Mobile Initialization 事件

jQuery Mobile Initialization 事件主要包括 pagebeforecreate 事件、pagecreate 事件和 pageinit 事件，它们的区别如下。

(1) pagebeforecreate 事件：发生在页面 DOM 加载后，正在初始化时。语法格式如下：

```
$(document).on("pagebeforecreate",function(){
   // 程序语句
});
```

(2) pagecreate 事件：发生在页面 DOM 加载完成，初始化也完成时。语法格式如下：



```
$(document).on("pagecreate",function(){
   // 程序语句
});
```

(3) pageinit 事件：发生在页面初始化完成以后。语法格式如下：

```
$(document).on("pageinit",function(){
   // 程序语句
});
```

下面通过一个综合案例来学习上面 3 个事件触发的时机。

【例 19.1】 初始化事件的触发(实例文件：ch19\19.1.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("pagebeforecreate",function(){
   alert("注意：pagebeforecreate 事件开始触发");
});
$(document).on("pagecreate",function(){
  alert("注意：pagecreate 事件开始触发");
});
$(document).on("pageinit",function(){
   alert("注意：pageinit 事件开始触发");
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>几回花下坐吹箫，银汉红墙入望遥。</p>
    <a href="#second">下一页</a>
  </div>
  <div data-role="footer">
    <h1>清代诗人</h1>
  </div>
</div>
<div data-role="page" id="second">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>似此星辰非昨夜，为谁风露立中宵。</p>
    <a href="#first">上一页</a>
  </div>
```

```
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中预览程序的效果，各个事件的执行顺序如图 19-1 所示。三次单击“确认”按钮后，结果如图 19-2 所示。

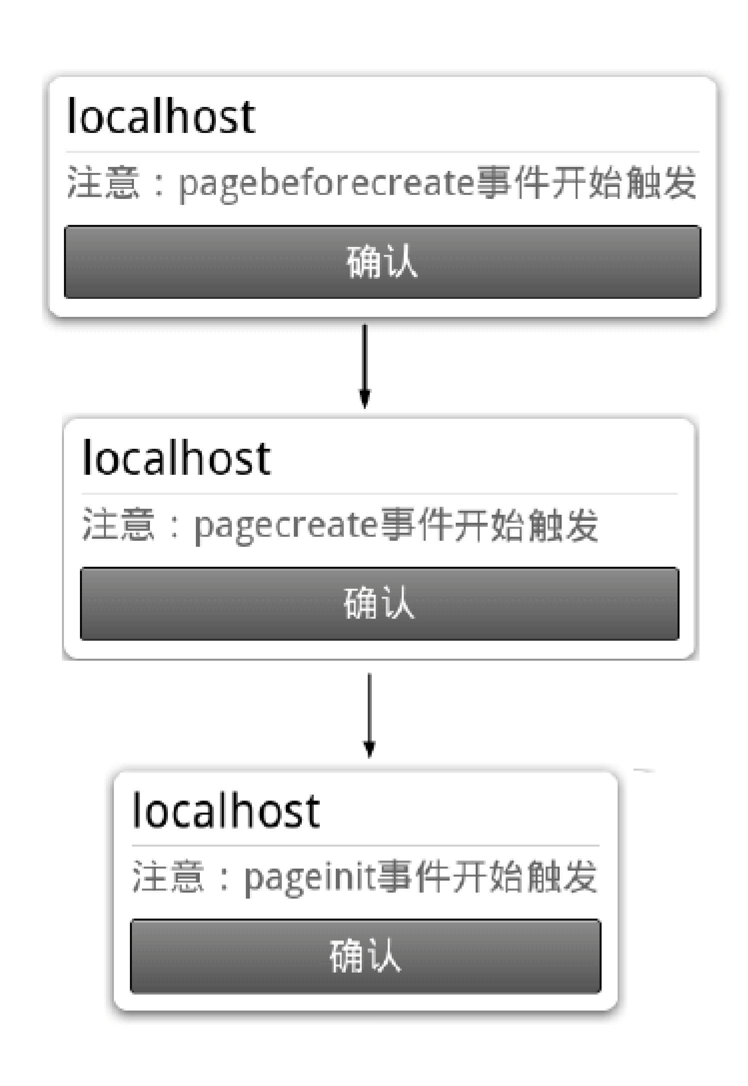


图 19-1　初始化事件



图 19-2　页面最终效果

19.1.2　案例 2——外部页面加载事件

外部页面加载时，最常见的加载事件如下。

1. pagebeforeload 事件

pagebeforeload 事件在外部页面加载前触发。语法格式如下：

```
<script>
$(document).on("pagebeforeload",function(){
  alert("有外部文件将要被加载);
});
</script>
```

2. pageload 事件

当页面加载成功时，触发 pageload 事件。语法格式如下：

```
<script>
$(document).on("pageload",function(event,data){
  alert("pageload 事件触发!\nURL: " + data.url);
});
</script>
```

pageload 事件函数的参数含义如下。

(1) event：任何 jQuery 的事件属性，如 event.type、event.pageX 和 target 等。

(2) data：包含以下属性。

① url：页面的 url 地址，是字符串类型。

② absUrl：绝对地址，是字符串类型。

③ dataUrl：地址栏 URL，是字符串类型。

④ options：$.mobile.loadPage()指定的选项，是对象类型。

⑤ xhr：XMLHttpRequest 对象，是对象类型。

⑥ textStatus：对象状态或空值，返回状态。

3. pageloadfailed 事件

如果页面载入失败，触发 pageloadfailed 事件。默认地，将显示 "Error Loading Page" 消息。语法格式如下：

```
$(document).on("pageloadfailed",function(event,data){
  alert("抱歉，被请求页面不存在。");
});
</script>
```

下面通过一个例子来理解上述事件触发时机。

【例 19.2】 触发 pageloadfailed 事件(实例文件：ch19\19.2.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("pageload",function(event,data){
alert("pageload 事件触发!\nURL: " + data.url);
});
$(document).on("pageloadfailed",function(){
  alert("抱歉，被请求页面不存在。");
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
    <p>众鸟高飞尽，孤云独去闲。相看两不厌，只有敬亭山。</p>
    <a href="123.html">下一页</a>
  </div>
  <div data-role="footer">
```

```
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-3 所示。单击“下一页”超链接，结果如图 19-4 所示。

古诗欣赏
众鸟高飞尽，孤云独去闲。
相看两不厌，只有敬亭山。
下一页
经典诗词

图 19-3 程序预览效果



图 19-4 触发 pageloadfailed 事件

19.1.3 案例 3——页面过渡事件

在 jQuery Mobile 中，在当前页面过渡到下一页时，会触发以下几个事件。

(1) pagebeforeshow 事件：在当前页面触发，在过渡动画开始前。

(2) pageshow 事件：在当前页面触发，在过渡动画完成后。

(3) pagebeforehide 事件：在下一页触发，在过渡动画开始前。

(4) pagehide 事件：在下一页触发，过渡动画完成后。

下面通过一个案例来学习页面过渡事件的触发时机。

【例 19.3】 页面过渡事件的触发(实例文件：ch19\19.3.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("pagebeforeshow","#second",function(){
  alert("触发 pagebeforeshow 事件，下一页即将显示");
});
$(document).on("pageshow","#second",function(){
  alert("触发 pageshow 事件，现在显示下一页");
});
$(document).on("pagebeforehide","#second",function(){
  alert("触发 pagebeforehide 事件，下一页即将隐藏");
});
```

```
$(document).on("pagehide","#second",function(){
  alert("触发 pagehide 事件，现在隐藏下一页");
});</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
    <p>青青园中葵，朝露待日晞。阳春布德泽，万物生光辉。</p>
    <a href="#second">下一页</a>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
  </div>

<div data-role="page" id="second">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
    <p>众鸟高飞尽，孤云独去闲。相看两不厌，只有敬亭山。</p>
    <a href="#first">上一页</a>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-5 所示。单击“下一页”超链接，事件触发顺序如图 19-6 所示。

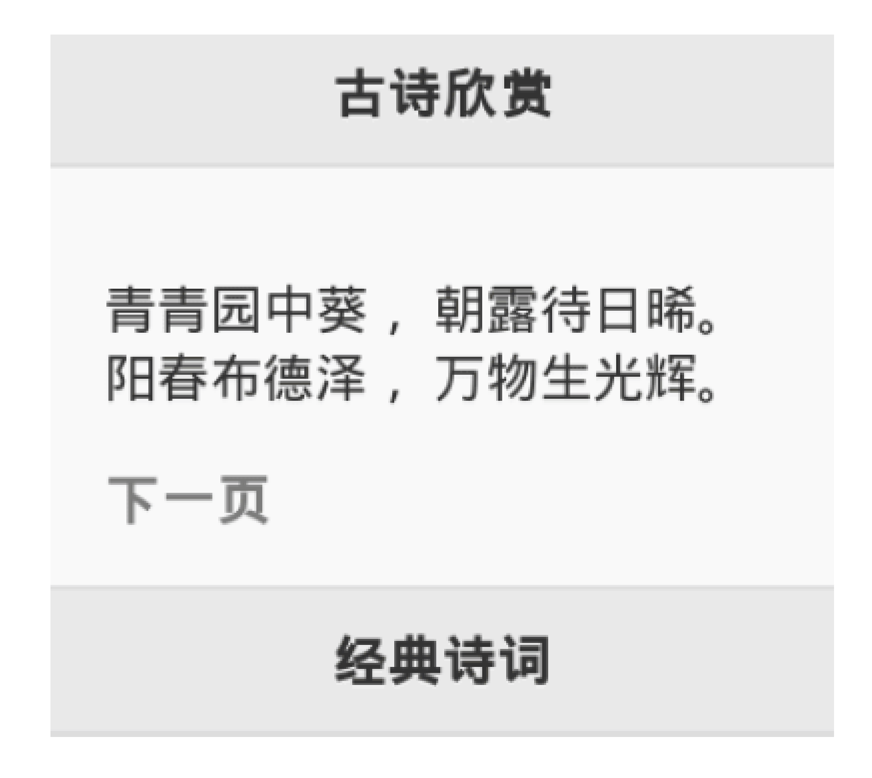


图 19-5　程序预览效果



图 19-6　当前页面触发事件顺序

单击“确认”按钮，进入下一页中，如图 19-7 所示。单击“上一页”超链接，事件触发顺序如图 19-8 所示。

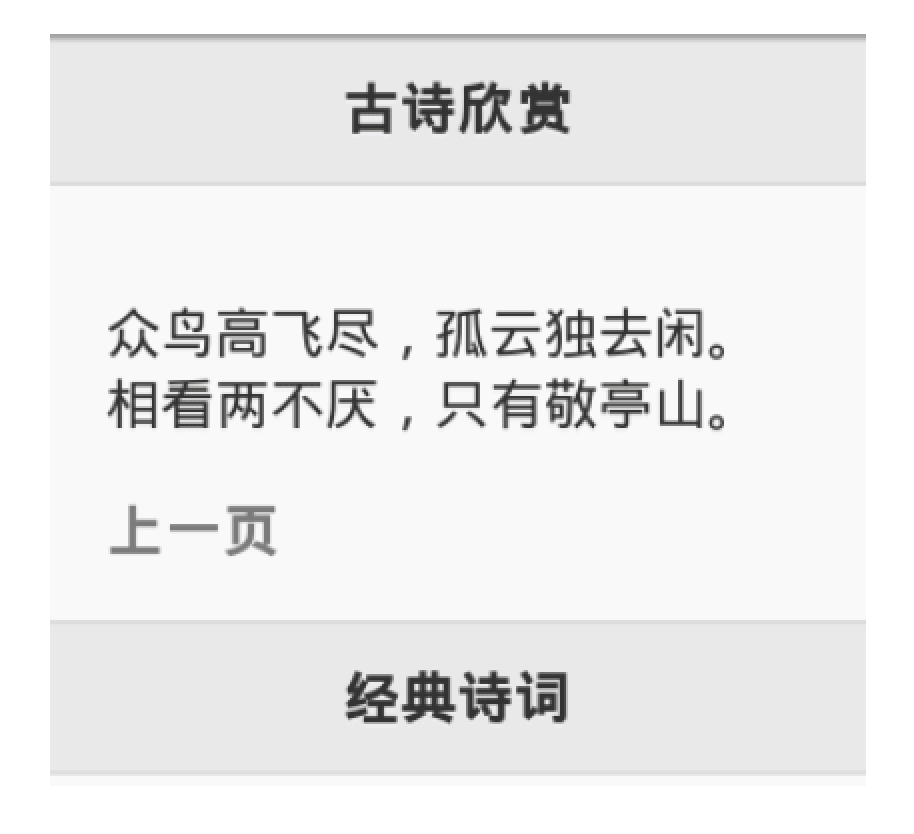


图 19-7　程序预览效果

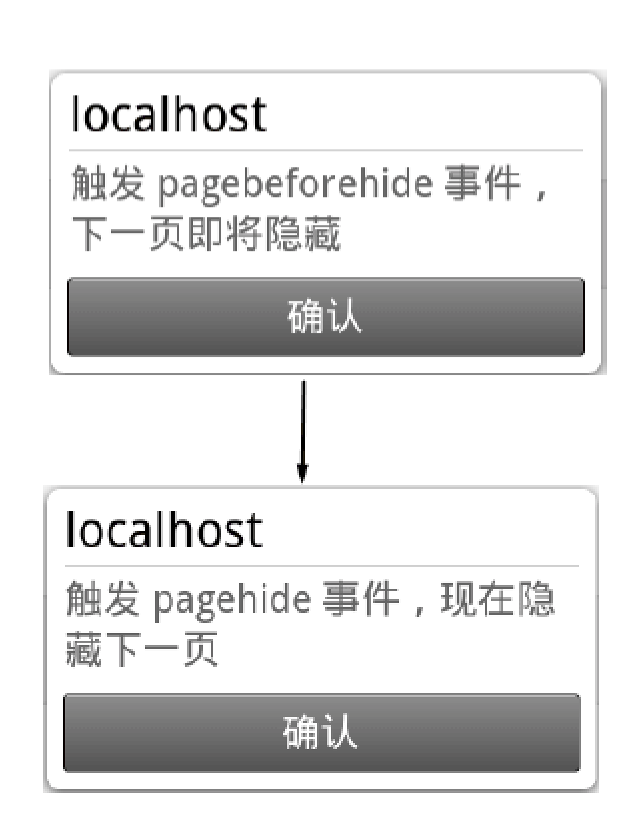


图 19-8　下一页触发事件顺序

19.2　触 摸 事 件

针对移动端浏览器提供了触摸事件，表示当用户触摸屏幕时触发的事件，包括点击事件和滑动事件。

19.2.1　案例 4——点击事件

点击事件包括 tap 事件和 taphold 事件。下面详细介绍它们的用法和区别。

1. tap 事件

当用户点击页面上的元素时，会触发点击(tap)事件，语法格式如下：

```
$("p").on("tap",function(){
  $(this).hide();
});
```

上述代码作用是点击 p 组件后，将会将该组件隐藏。

下面通过一个案例来讲解点击事件的使用方法。

【例 19.4】 点击事件的使用(实例文件：ch19\19.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$("div").on("tap",function(){
  $(this).css("color","green");
});
```

```
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
        <p>黄师塔前江水东，春光懒困倚微风。桃花一簇开无主，可爱深红爱浅红。</p>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-9 所示。在页面中诗词上面点击，即可发现 div 块内文字的颜色变成了绿色，如图 19-10 所示。

古诗欣赏

黄师塔前江水东，春光懒困倚微风。
桃花一簇开无主，可爱深红爱浅红。

经典诗词

图 19-9　程序预览效果

古诗欣赏

黄师塔前江水东，春光懒困倚微风。
桃花一簇开无主，可爱深红爱浅红。

经典诗词

图 19-10　触发 tap 事件

2. taphold

如果点击页面并按住不放，则会触发 taphold 事件。语法格式如下：

```
$("p").on("taphold",function(){
  $(this).hide();
});
```

在默认情况下，按住不放 750ms 之后触发 taphold 事件。用户也可以修改这个时间的长短。语法格式如下：

```
$(document).on("mobileinit",function(){
  $.event.special.tap.tapholdThreshold=5000;
});
```

修改后需要按住 5 秒以后才会触发 taphold 事件。

【例 19.5】 触发 taphold 事件(实例文件：ch19\19.5.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("mobileinit",function(){
  $.event.special.tap.tapholdThreshold=1000
});
$(function(){
  $("img").on("taphold",function(){
   $(this).hide();
});
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>可爱宠物</h1>
  </div>
  <div data-role="content" class="content">
<img src=19.1.jpg > <br>
        <p>按住图片 1 秒后隐藏图片哦! </p>
  </div>
  <div data-role="footer">
    <h1>动物天地</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-11 所示。点击图片 1 秒后，即可发现图片被隐藏了，如图 19-12 所示。



图 19-11　程序预览效果



图 19-12　触发 taphold 事件

19.2.2　案例 5——滑动事件

滑动事件是在用户 1 秒内水平拖曳大于 30px，或者纵向拖曳小于 20px 的事件发生时触发

的事件。滑动事件使用 swipe 语法来捕捉。语法格式如下：

```
$("p").on("swipe",function(){
  $("span").text("滑动检测!");
});
```

上述语法是捕捉 p 组件的滑动事件，并将消息显示在 span 组件中。

向左滑动事件在用户向左拖动元素大于 30px 时触发，使用 swipeleft 语法来捕捉、语法格式如下：

```
$("p").on("swipeleft",function(){
  $("span").text("向左滑动检测!");
});
```

向右滑动事件在用户向右拖动元素大于 30px 时触发，使用 swiperight 语法来捕捉。语法格式如下：

```
$("p").on("swiperight,function(){
  $("span").text("向右滑动检测!");
});
```

下面以向右滑动事件为例进行讲解。

【例 19.6】 触发向右滑动事件(实例文件：ch19\19.6.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("pagecreate","#first",function(){
   $("img").on("swiperight",function(){
    alert("干嘛向右滑动我!!");
   });
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>可爱宠物</h1>
  </div>
  <div data-role="content" class="content">
<img src=19.2.jpg > <br>
         <p>向右滑动图片查看效果</p>
  </div>
  <div data-role="footer">
    <h1>动物天地</h1>
  </div>
</div>
```

```
</body>
</html>
```

在模拟器中的预览效果如图 19-13 所示。向右滑动图片，效果如图 19-14 所示。

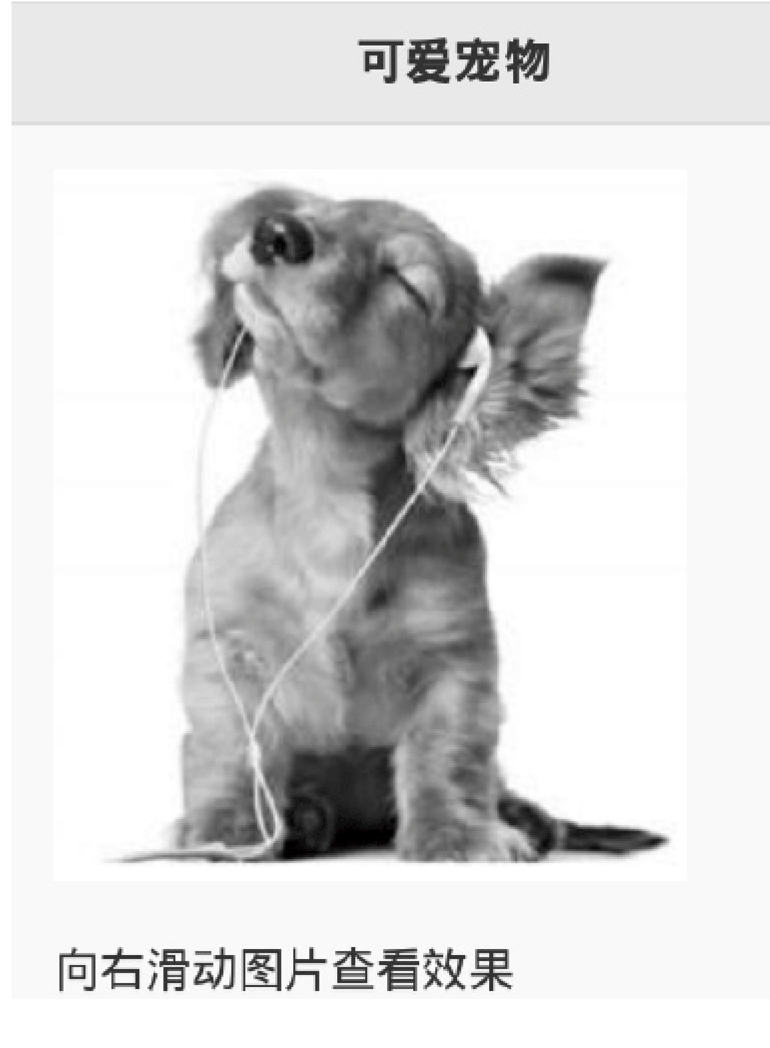


图 19-13　程序预览效果



图 19-14　触发向右滑动事件

19.3　案例 6——滚屏事件

jQuery Mobile 提供了两种滚屏事件，分别是滚屏开始时触发 Scrollstart 事件和滚屏结束时触发 Scrollstop 事件。

1. Scrollstart 事件

scrollstart 事件是在用户开始滚动页面时触发。语法格式如下：

```
 $(document).on("scrollstart",function(){
   alert("屏幕开始滚动了!");
 });
```

下面通过一个案例来理解 Scrollstart 事件。

【例 19.7】 触发 Scrollstart 事件(实例文件：ch19\19.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("pagecreate","#first",function(){
```

```
  $(document).on("scrollstart",function(){
    alert("屏幕开始滚动了！");
  });
  });
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
    <p>西施越溪女，出自苎萝山。</p>
    <p>秀色掩今古，荷花羞玉颜。</p>
    <p>浣纱弄碧水，自与清波闲。</p>
    <p>皓齿信难开，沉吟碧云间。</p>
    <p>勾践徵绝艳，扬蛾入吴关。</p>
    <p>提携馆娃宫，杳渺讵可攀。</p>
    <p>一破夫差国，千秋竟不还。</p>
    <p>西施越溪女，出自苎萝山。</p>
    <p>秀色掩今古，荷花羞玉颜。</p>
    <p>浣纱弄碧水，自与清波闲。</p>
    <p>皓齿信难开，沉吟碧云间。</p>
    <p>勾践徵绝艳，扬蛾入吴关。</p>
    <p>提携馆娃宫，杳渺讵可攀。</p>
    <p>一破夫差国，千秋竟不还。</p>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-15 所示。向上滚动屏幕，效果如图 19-16 所示。



图 19-15　程序预览效果



图 19-16　触发 Scrollstart 事件

2. Scrollstop 事件

Scrollstop 事件是在用户停止滚动页面时触发。语法格式如下：

```
$(document).on("scrollstop",function(){
 alert("停止滚动!");
});
```

下面通过一个案例来理解 Scrollstop 事件。

【例 19.8】 触发 Scrollstop 事件(实例文件：ch19\19.8.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
 <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script>
$(document).on("pagecreate","#first",function(){
  $(document).on("scrollstop",function(){
    alert("屏幕已经停止滚动了!");
  });
});
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
    <p>噫吁嚱，危乎高哉！</p>
    <p>蜀道之难，难于上青天！</p>
    <p>蚕丛及鱼凫，开国何茫然！</p>
    <p>尔来四万八千岁，不与秦塞通人烟。</p>
    <p>西当太白有鸟道，可以横绝峨嵋巅。</p>
    <p>地崩山摧壮士死，然后天梯石栈方钩连。</p>
    <p>上有六龙回日之高标，下有冲波逆折之回川。</p>
    <p>黄鹤之飞尚不得过，猿猱欲度愁攀援。</p>
    <p>青泥何盘盘，百步九折萦岩峦。</p>
    <p>扪参历井仰胁息，以手抚膺坐长叹。</p>
    <p>问君西游何时还？畏途巉岩不可攀。</p>
    <p>但见悲鸟号古木，雄飞从雌绕林间。</p>
    <p>又闻子规啼夜月，愁空山。</p>
    <p>蜀道之难，难于上青天，使人听此凋朱颜。</p>
    <p>连峰去天不盈尺，枯松倒挂倚绝壁。</p>
    <p>飞湍瀑流争喧豗，砯崖转石万壑雷。</p>
    <p>其险也若此，嗟尔远道之人，胡为乎来哉。</p>
    <p>剑阁峥嵘而崔嵬，一夫当关，万夫莫开。</p>
    <p>所守或匪亲，化为狼与豺。</p>
    <p>朝避猛虎，夕避长蛇，磨牙吮血，杀人如麻。</p>
    <p>锦城虽云乐，不如早还家。</p>
```

```
    <p>蜀道之难，难于上青天，侧身西望长咨嗟。</p>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-17 所示。向上滚动屏幕，停止后效果如图 19-18 所示。



图 19-17　程序预览效果



图 19-18　触发 Scrollstop 事件

19.4　案例 7——定位事件

当移动设备水平或垂直翻转时触发定位事件，也就是常说的方向改变(orientationchange)事件。

在使用定位事件时，请将 orientationchange 事件绑定到 window 对象上。语法格式如下：

```
$(window).on("orientationchange",function(event){
alert("设备的方向改变为"+ event.orientation);
});
```

这里的 event 对象用来接收 orientation 属性值，用 event.orientation 返回的是设备是水平还是垂直，类型为字符串。如果是横向，返回值为 landscape；如果是纵向，返回值为 portrait。

下面通过一个案例来理解 orientationchange 事件。

【例 19.9】 定位事件的触发(实例文件：ch19\19.9.html)。

```
<!DOCTYPE html>
```

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script type="text/javascript">
    $(document).on("pageinit",function(event){
        $( window ).on( "orientationchange", function( event ) {
         if(event.orientation == "landscape")
           $( "#orientation" ).text( "现在是水平模式!" ).css({"background-
color":"yellow","font-size":"300%"});
         if(event.orientation == "portrait")
            $( "#orientation" ).text( "现在是垂直模式!" ).css({"background-
color":"green","font-size":"200%"});
         });
        })
</script>
</head>
<body>
<div data-role="page" id="first">
  <div data-role="header">
    <h1>古诗欣赏</h1>
  </div>
  <div data-role="content" class="content">
<span id="orientation"></span><br>
<p>燕草如碧丝，秦桑低绿枝。当君怀归日，是妾断肠时。春风不相识，何事入罗帏</p>
  </div>
  <div data-role="footer">
    <h1>经典诗词</h1>
  </div>
</div>
</body>
</html>
```

在模拟器中的预览效果如图 19-19 所示。单击模拟器上的方向改变按钮，此时方向改变为水平方向，效果如图 19-20 所示。



图 19-19　程序预览效果



图 19-20　设备是水平方向



再次单击模拟器上的方向改变按钮，此时方向改变为垂直方向，效果如图 19-21 所示。



图 19-21 设备是垂直方向

19.5 高 手 解 惑

疑问 1：绑定事件的方法 on()和 one()的区别？

答：绑定事件的方法 on()和 one()的作用相似，它们唯一的区别在于 one()只能执行一次。例如，当在按钮上绑定单击鼠标事件时，on()方法的程序如下：

```
<script>
$(document).on('click',function(){
   alert("这是使用 on()方法绑定的事件")
});
</script>
```

疑问 2：如何在设备方向改变时获取移动设备的高度和宽度？

答：如果设备方向改变时要获取移动设备的高度和宽度，可以绑定 resize 事件。该事件在页面大小改变时将触发，语法如下：

```
$(window).on("resize",function(){
  var win= $(this);    //this 指的是 window
  alert("宽度为"+win.width()+"高度为"+ win.height());
});
```



在第 14 章中介绍了如何利用 localStorage 实现本地存储。实际上，除了 sessionStorage 和 localStorage 外，HTML 5 还支持通过本地数据库 Web SQL Database 进行本地数据存储。为此，本章介绍本地数据库的数据存储和读取技术。



20.1　Web SQL Database 概述

Web SQL Database 是关系型数据库系统，使用 SQLLite 语法访问数据库，支持大部分浏览器，该数据库多集中在嵌入式设备上。

Web SQL Database 数据库中定义的 3 个核心方法如下。

(1) openDatabase：这个方法使用现有数据库或新建数据库来创建数据库对象。

(2) executeSql：这个方法用于执行 SQL 查询。

(3) transaction：这个方法允许用户根据情况控制事务提交或回滚。

在 Web SQL Database 中，用户可以打开数据库并进行数据的新增、读取、更新、删除等操作。操作数据的基本流程如下。

(1) 创建数据库。

(2) 创建交易(transaction)。

(3) 执行 SQL 语法。

(4) 获取 SQL 语句执行的结果。

20.2　数据库的基本操作

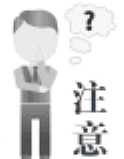
数据库的基本操作如下。

1. 创建数据库

使用 openDatabase 方法打开一个已经存在的数据库；如果数据库不存在，使用此方法将会创建一个新数据库。打开或创建一个数据库的代码如下：

```
var db = openDatabase('mydb', '1.1', ' 第一个数据库', 200000);
```

上述代码的括号中设置了 5 个参数，其意义分别为：数据库名称、版本号、文字说明、数据库的大小和创建回滚。



如果数据库已经创建了，第五个参数将会调用此回滚操作。如果省略此参数，则仍将创建正确的数据库。

以上代码的意义为：创建了一个数据库对象 db，名称是 mydb，版本编号为 1.1。db 还带有描述信息和大概的大小值。用户代理(user agent)可使用这个描述与用户进行交流，说明数据库是用来做什么的。利用代码中提供的大小值，用户代理可以为内容留出足够的存储。如果需要，这个大小是可以改变的，所以没有必要预先假设允许用户使用多少空间。

2. 创建交易

创建交易时使用 database.transaction()函数，语法格式如下：

```
db.transaction(function(tx){
//执行访问数据库的语句
```

```
});
```

该函数使用 function(tx)作为参数，执行访问数据库的具体操作。

3. 执行 SQL 语句

通过 executeSql 方法执行 SQL 语句，从而对数据库进行操作，代码如下：

```
tx.executeSql(sqlQuery,[value1,value2..],dataHandler,errorHandler)
```

executeSql 方法有 4 个参数，作用分别如下。

(1) sqlQuery：需要具体执行的 sql 语句，可以是 create、select、update、delete。

(2) [value1,value2..]：SQL 语句中所有使用到的参数的数组，在 executeSql 方法中，将 SQL 语句中所要使用的参数先用“?”代替，然后依次将这些参数组成数组放在第二个参数中。

(3) dataHandler：执行成功是调用的回调函数，通过该函数可以获得查询结果集。

(4) errorHandler：执行失败时调用的回调函数。

4. 获取 SQL 语句执行的结果

当 SQL 语句执行成功后，就可以使用循环语句来获取执行的结果，代码如下：

```
for (var a=0; a<result.rows.length; a++){
    item = result.rows.item(a);
    $("div").html(item["name"] +"<br>");
}
```

result.rows 表示结果数据，result.rows.length 表示数据共有几条，然后通过 result.rows.item(a)获取每条数据。

20.3　数据表的基本操作

创建数据表的语句为 CREATE　TABLE，语法格式如下：

```
CREATE  TABLE <表名>
(
    字段名 1 数据类型 [约束条件],
    字段名 2 数据类型 [约束条件],
    …
);
```

使用 CREATE TABLE 创建表时，必须指定以下信息。

(1) 要创建的表的名称，不区分大小写，不能使用 SQL 语言中的关键字，如 DROP、ALTER、INSERT 等。

(2) 数据表中每一个列(字段)的名称和数据类型，如果创建多个列，要用逗号隔开。

例如，创建员工表 tb_emp1，结构如表 20-1 所示。

表 20-1　tb_emp1 表结构

字段名称	数据类型	备　注
id	int	员工编号
name	char(10)	员工名称
introduction	varchar(300)	员工简介

创建 tb_emp1 表，SQL 语句为：

```
CREATE TABLE tb emp1
(
   id      int PRIMARY KEY,
   name   char(10),
   introduction  varchar(300)
);
```

其中 PRIMARY KEY 约束条件定义 id 字段为主键。如果数据表已经存在，则上述创建命令将会报错，此时可以加入 if not exists 命令先进行条件判断。

下面通过一个综合案例来学习。

【例 20.1】 创建数据表(实例文件：ch20\20.1.html)。

```
<!DOCTYPE html>
<html>
  <head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script type="text/javascript">
$(function () {
    //打开数据库
    var dbSize=2*1024*1024;
     db = openDatabase('mytestDB', '', '', dbSize);
     //创建数据表
     db.transaction(function(tx){
        tx.executeSql("CREATE TABLE IF NOT EXISTS student (id integer
        PRIMARY KEY, name char(10),introduction
        varchar(300) )",[],onSuccess,onError);
    });
    function onSuccess(tx, results)
    {
      $("div").html("数据库打开成功!")
    }
    function onError(e)
    {
      $("div").html("数据库打开错误:"+e)
    }

})
</script>
</head>
<body>
    <div id="message"></div>
```

```
</body>
</html>
```

执行结果如图 20-1 所示。

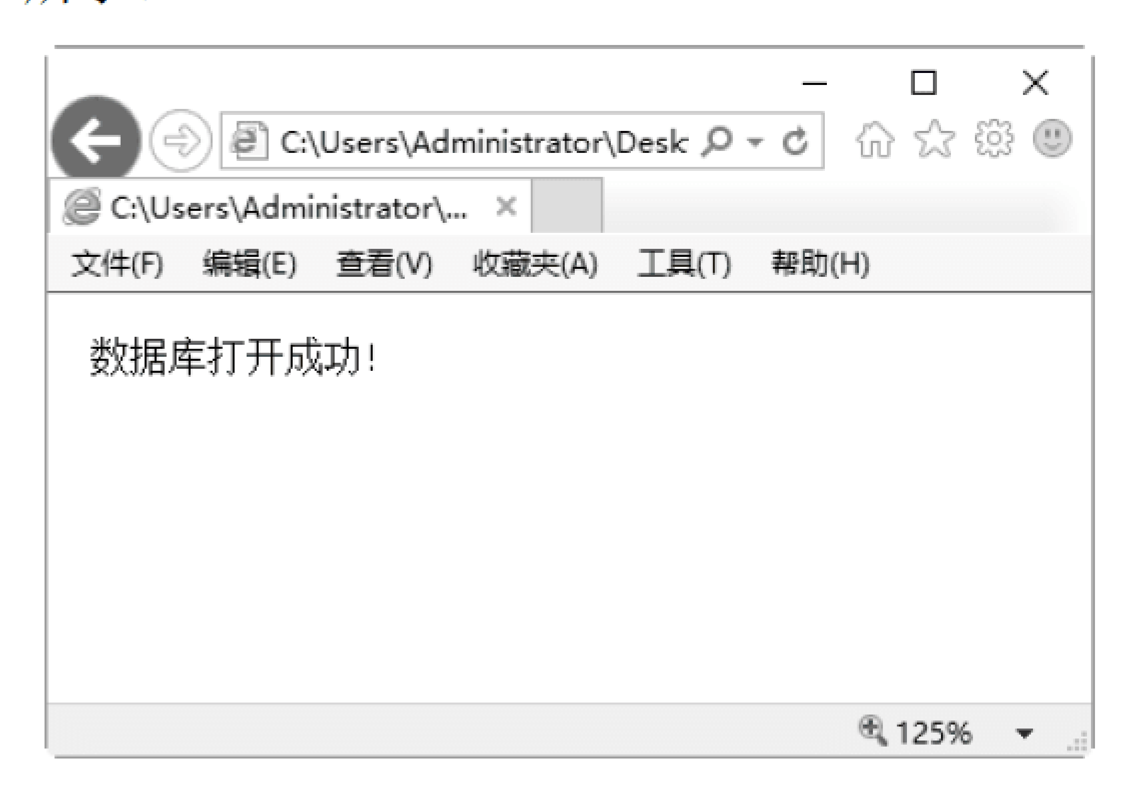


图 20-1　程序运行结果

20.4　数据的基本操作

数据表创建完成后，即可对数据进行添加、更新、查询、删除等操作。

1. 添加数据

添加数据的语法规则如下。

使用基本的 INSERT 语句插入数据，要求指定表的名称和插入到新记录中的值。基本语法格式如下：

```
INSERT INTO table_name (column_list) VALUES (value_list);
```

table_name 指定要插入数据的表名；column_list 指定要插入数据的哪些列；value_list 指定每个列对应插入的数据。注意，使用该语句时字段列和数据值的数量必须相同。

例如向数据表 student 添加一条数据，语句如下：

```
INSERT INTO student (id ,name, introduction) VALUES (1,'lili', 'she is a
good student');
```

在添加字符串时，必须使用单引号。

2. 更新数据

表中有数据之后，接下来可以对数据进行更新操作，MySQL 中使用 UPDATE 语句更新表中的记录，可以更新特定的行或者同时更新所有的行。基本语法格式如下：

```
UPDATE table name
SET column name1 = value1,column name2=value2,…,column namen=valuen
WHERE (condition);
```

“column_name1,column_name2,…,column_namen”为指定更新的字段的名称；“value1,

value2,…,valuen”为相对应的指定字段的更新值；condition 指定更新的记录需要满足的条件。更新多个列时，每个“列-值”对之间用逗号隔开，最后一列之后不需要逗号。

例如，在表 student 中，更新 id 值为 1 的记录，将 name 字段值改为 LiMing，语句如下：

```
UPDATE student SET name= 'LiMing' WHERE id = 1;
```

3. 查询数据

查询数据使用 SELECT 命令，语法格式如下：

```
SELECT value1, value2 FROM table_name WHERE (condition);
```

例如，在表 student 中，查询 name 字段值为 LiMing 的记录，语句如下：

```
SELECT id ,name, introduction FROM student WHERE name= 'LiMing';
```

4. 删除数据

从数据表中删除数据使用 DELETE 语句，DELETE 语句允许 WHERE 子句指定删除条件。DELETE 语句基本语法格式如下：

```
DELETE FROM table_name [WHERE <condition>];
```

table_name 指定要执行删除操作的表；“[WHERE <condition>]”为可选参数，指定删除条件，如果没有 WHERE 子句，DELETE 语句将删除表中的所有记录。

例如，表 student 中，删除 name 字段值为 LiMing 的记录，语句如下：

```
DELETE FROM student WHERE name= 'LiMing';
```

20.5 综合案例——Web SQL Database 的综合操作技能

下面模拟个人信息数据库，然后使用 Web SQL Database 数据库实现数据库创建和数据新增、查看和删除等操作。

【例 20.2】 数据库的综合操作(实例文件：ch20\20.2.html)。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
<style>
table{border-collapse:collapse;}
td{border:1px solid #0000cc;padding:5px}
#message{color:#ff0000}
</style>
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script type="text/javascript">
$(function () {
    //打开数据库
    var dbSize=2*1024*1024;
     db = openDatabase('myDB', '', '', dbSize);
```

```
    db.transaction(function(tx){
        //创建数据表
        tx.executeSql("CREATE TABLE IF NOT EXISTS person (id integer
          PRIMARY KEY,name char(10),introduction varchar(200))");
        showAll();
    });

    $( "button" ).click(function () {
        var name=$("#name").val();
        var introduction=$("#introduction").val();
        if(name=="" || introduction==""){
            $("#message").html("**请输入姓名和简介**");
            return false;
        }

        db.transaction(function(tx){
                //新增数据
                tx.executeSql("INSERT INTO person(name,introduction)
                  values(?,?)",[name,introduction],function(tx, result){
                     $("#message").html("新增数据完成!")
                     showAll();
                },function(e){
                     $("#message").html("新增数据错误:"+e.message)
                });
        });
    })

    $("#showData").on('click', ".delItem", function() {
        var delid=$(this).prop("id");
        db.transaction(function(tx){
                //删除数据
                var delstr="DELETE FROM person WHERE id=?";
                tx.executeSql(delstr,[delid],function(tx, result){
                     $("#message").html("删除数据完成!")
                    showAll();
                },function(e){
                     $("#message").html("删除数据错误:"+e.errorCode);
                });
        });
    })
    function showAll(){
        $("#showData").html("");
        db.transaction(function(tx){
            //显示 person 数据表全部数据
            tx.executeSql("SELECT id,name,introduction FROM person",[],
              function(tx, result){
                if(result.rows.length>0){
                    var str="现有数据: <br><table><tr><td>id</td><td>姓名
                      </id><td>简介</id><td> </id></tr>";
                    for(var i = 0; i < result.rows.length; i++){
                        item = result.rows.item(i);
                        str+="<tr><td>"+item["id"] + "</td><td>" +
item["name"] + "</td><td>" + item["introduction"] + "</td><td><input
type='button' id='"+item["id"]+"' class='delItem' value='删除'></td></tr>";
```

```
                    }
                    str+="</table>";
                    $("#showData").html(str);
                }
            },function(e){
                $("#message").html("SELECT 语法出错了!"+e.message)
            });
        });
    }

})
</script>
</head>
<body>
<h3>数据新增与删除</h3>
请输入姓名和简介:
<table>
<tr>
    <td>姓名: </td>
    <td><input type="text" id="name"></td>
</tr>
<tr>
    <td>简介: </td>
    <td><input type="text" id="introduction"></td>
</tr>
</table>
<button id='new'>发送</button>
<p>
<div id="message"></div>

<div id="showData"></div>
</body>
</html>
```

程序运行结果如图 20-2 所示，输入姓名和简介后，单击“提交”按钮，即可看到新提交的数据，单击“删除”按钮，即可删除选中的数据。

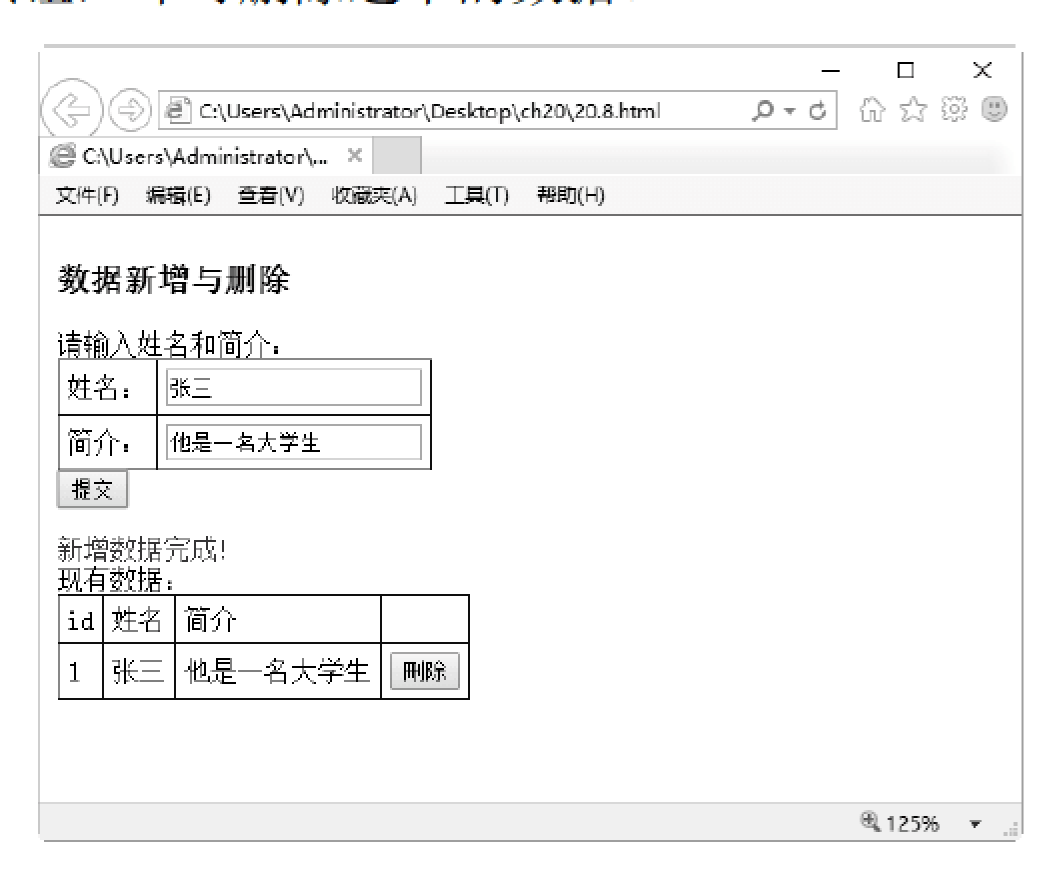


图 20-2　程序运行结果



20.6 高 手 解 惑

疑问 1：如何检查连接数据库是否成功？

答：为了检测创建的连接是否成功，可以检查那个数据库对象是否为 null：

```
if(!db)
alert("数据库连接失败");
```

绝不可以假设该连接已经成功建立，即使过去对于某个用户它是成功的。为什么一个连接会失败，存在多个原因。也许用户代理出于安全原因拒绝你的访问，也许设备存储有限。面对活跃而快速进化的潜在用户代理，对用户的机器、软件及其能力做出假设是非常不明智的行为。

疑问 2：无法成功插入数据怎么办？

答：在插入数据时，需要保证字段和数据值的数量一样，而且数据类型也要一一对应，否则会插入数据失败。特别是插入字符串时，一定要使用单引号。

第 V 篇

综合案例实战

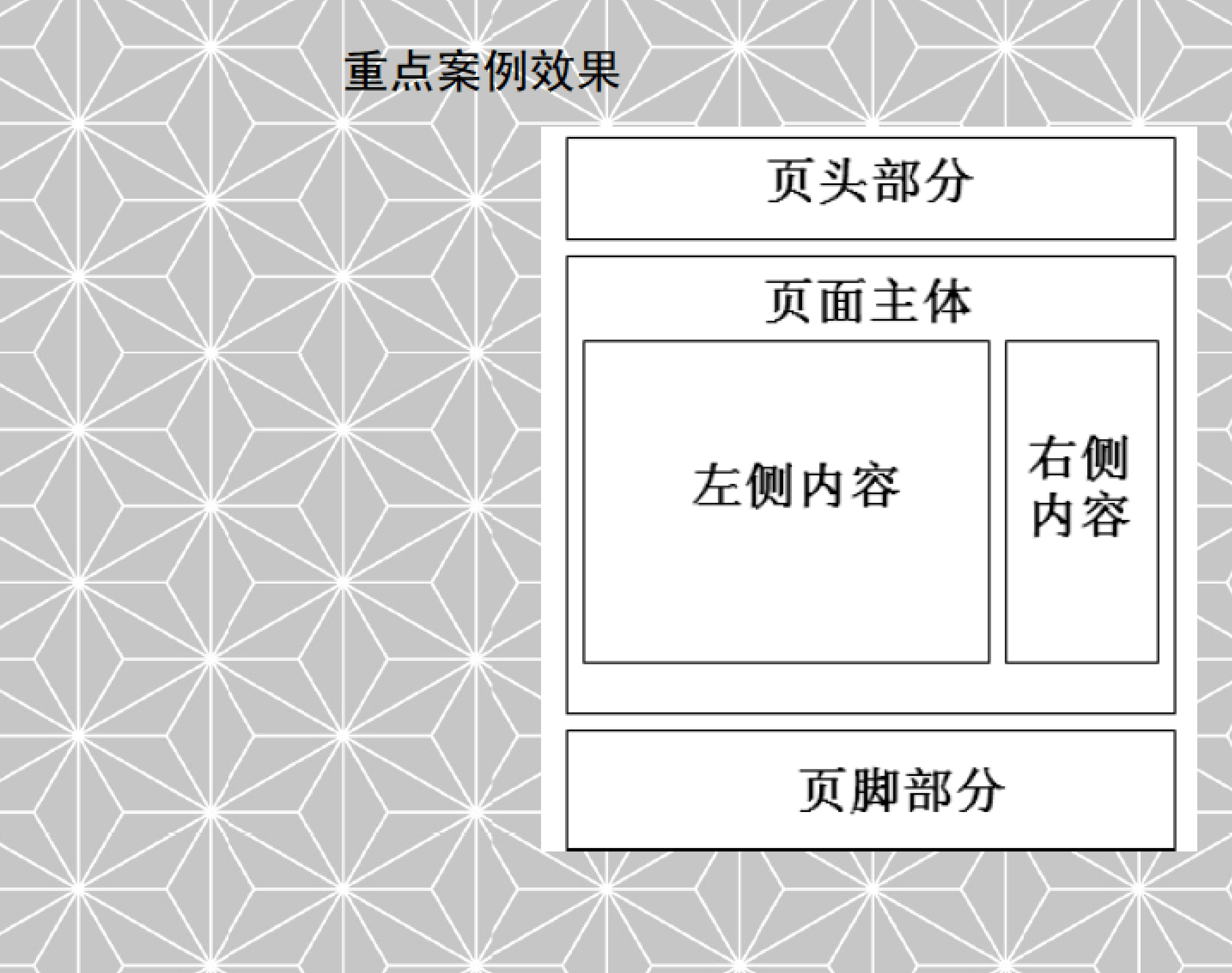
- 第 21 章　制作休闲娱乐类网页
- 第 22 章　制作企业门户类网页
- 第 23 章　制作电子商务类网页
- 第 24 章　开发连锁酒店订购系统



第 21 章 制作休闲娱乐类网页

休闲娱乐类的网页种类很多，如聊天交友、星座运程、游戏视频等。本章主要以视频类网页为例进行介绍。视频类网页主要包含视频搜索、播放、评价、上传等内容。此类网站都会容纳各种类型的视频信息，让浏览者轻松地找到自己需要的视频。

重点案例效果





21.1 整 体 布 局

本实例以简单的视频播放页面为例来演示视频网站的制作方法。网页内容应包括：头部、导航菜单栏、检索条、视频播放及评价、热门视频推荐等内容。使用浏览器浏览其完成后的效果，如图 21-1 所示。



图 21-1　视频播放网页效果

21.1.1　设计分析

作为一个视频播放网页，页面应简单、明了，给人以清晰的感觉。整体设计各部分内容介绍如下。

(1) 页头部分主要放置导航菜单和网站 Logo 信息等，其 Logo 可以是一张图片或者文本信息等。

(2) 页头下方应是搜索模块，用于帮助浏览者快速检索视频。

(3) 页面主体左侧是视频播放及评价，考虑到视频播放效果，左侧主题部分至少要占整个页面 2/3 的宽度，另外要为视频增加信息描述内容。

(4) 页面主体右侧是热门视频推荐模块、当前热门视频和根据当前播放的视频类型推荐的视频。

(5) 页面底部是一些快捷链接和网站备案信息。

21.1.2 排版架构

从图 21-1 所示的效果图可以看出，页面结构并不是太复杂，采用的是上中下结构，页面主体部分又嵌套了一个左右版式结构，其效果如图 21-2 所示。



图 21-2 网页架构

21.2 模 块 组 成

在制作网站的时候，可以将整个网站划分为三大模块，即上、中、下。框架实现代码如下：

```
<div id="main_block">                         //主体框架
  <div id="innerblock">                       //内部框架
      <div id="top_panel">                    //头部框架
      </div>
      <div id="contentpanel">                 //中间主体框架
      </div>
      <div id="ft_padd">                      //底部框架
      </div>
  </div>
</div>
```

以上框架结构比较粗糙，想要页面内容布局完美，需要更细致的框架结构。

1. 头部框架

头部框架实现代码如下：

```
<div id="top panel">
   <div class="tp navbg">             //导航栏模块框架
   </div>
   <div class="tp smlgrnbg">          //注册登录模块框架
   </div>
```

```
    <div class="tp barbg">              //搜索模块框架
    </div>
</div>
```

2. 中间主体框架

中间主体框架实现代码如下：

```
<div id="contentpanel">                  //中间主体框架
    <div id="lp padd">                   //中间左侧框架
        <div class="lp newvidpad" style="margin-top:10px;">      //评论模块框架
        </div>
    </div>
    <div id="rp padd">                   //中间右侧框架
        <div class="rp loginpad" style="padding-bottom:0px; border-bottom:
          none;">
        //右侧上部模块框架
        </div>
        <div class="rp loginpad" style="padding-bottom:0px; border-bottom:
          none;">
        //右侧下部模块框架
        </div>
    </div>
</div>
```



其中大部分框架参数中只有一个框架 ID 名，而部分框架中添加了其他参数，一般只有 ID 名的框架在 CSS 样式表中都有详细的框架属性信息。

3. 底部框架

底部框架实现代码如下：

```
 <div id="ft_padd">
    <div class="ftr lnks">   //底部快速链接模块框架
    </div>
</div>
```

21.3 制作步骤

网站制作要逐步完成。本实例中网页制作主要包括 7 个部分，详细制作方法介绍如下。

21.3.1 制作样式表

为了更好地实现网页效果，需要为网页制作 CSS 样式表。制作样式表的实现代码如下：

```
/* CSS Document */
body{
margin:0px; padding:0px;
font:11px/16px Arial, Helvetica, sans-serif;
background:#0C0D0D url(../images/bd_bg1px.jpg) repeat-x;
```

```
}
p{
margin:0px;
padding:0px;
}
img
{
border:0px;
}
a:hover
{
text-decoration:none;
}

#main block
{
margin:auto; width:1000px;
}
#innerblock
{
float:left; width:1000px;
}

#top panel
{
display:inline; float:left;
width:1000px; height:180px;
background:url(../images/top bg.jpg) no-repeat;
}
.logo
{
float:left; margin:40px 0 0 30px;
}
.tp navbg
{
 clear:left; float:left;
  width:590px; height:32px;
  display:inline;
  margin:26px 0 0 22px;
  }
.tp_navbg a
{
float:left; background:url(../images/tp inactivbg.jpg) no-repeat;
 width:104px; height:19px;
 padding:13px 0 0 0px; text-align:center;
 font:bold 11px Arial, Helvetica, sans-serif;
 color:#B8B8B4; text-decoration:none;
 }
.tp navbg a:hover
{
float:left; background:url(../images/tp activbg.jpg) no-repeat;
width:104px; height:19px; padding:13px 0 0 0px; text-align:center;
font:bold 11px Arial, Helvetica, sans-serif; color:#282C2C;
text-decoration:none;
```

```
}
.tp_smlgrnbg{
float:left; background:url(../images/tp smlgrnbg.jpg) no-repeat;
margin:34px 0 0 155px; width:160px; height:24px;
}
.tp_sign{float:left; margin:6px 0 0 19px;}
.tp txt{
float:left; margin:0px 0 0 0px;
font:11px/15px Arial; color:#FFFFFF;
text-decoration:none; display:inline;
}
.tp divi{
float:left; margin:0px 8px 0 8px;
font:11px/15px Arial; color:#FFFFFF;
display:inline;
}

.tp barbg
{
float:left; background:url(../images/tp barbg.jpg) repeat-x;
width:1000px; height:42px;
width:1000px;
}
.tp barip
{
float:left; width:370px;
 height:20px; margin:8px 0 0 173px;
 }
.tp_drp{
float:left; margin:8px 0 0 10px;
 width:100px; height:24px;
 }
.tp_search{
float:left;
margin:8px 0 0 10px;
 }
.tp_welcum{
float:left; margin:14px 0 0 80px;
font:11px Arial, Helvetica, sans-serif;
color:#2E3131; width:95px;
}

#contentpanel{
clear:left; float:left; width:1000px;
display:inline; margin-top:9px;
 padding-bottom:20px;
 }

#lp padd{
float:left; width:665px;
display:inline; margin:0 0 0 22px;
}
.lp shadebg{
 float:left; background:#0C0D0D url(../images/lp_shadebg.jpg) no-repeat;
```

```
  width:660px; height:144px;
  }
.lp watch{ float:left; margin-top:24px;}
.cp_watcxt{
float:left; margin:9px 0 0 7px;
 width:110px; font:11px/16px Arial, Helvetica, sans-serif;
 color:#A1A1A1;
 }
.cp smlpad{
float:left; width:200px;
display:inline;
}
.cp watchit{ float:left; margin:30px 0 0 7px;}
.lp uplad{ float:left; margin-top:24px;}
.lp newline{ float:left; margin:6px 0 0 0;}
.lp arro{ float:left; margin:55px 0 0 7px;}
.lp newvid1{ float:left; margin:10px 0 0 10px;}
.lp newvidarro{ clear:left; float:left; margin:13px 0 0 10px;}
.lp featimg1{ clear:left; float:left; margin:35px 0 0 17px;}
.lp featline{ clear:left; float:left; margin:28px 0 0 15px;}
.lp watmore{
float:left; display:inline;
margin:5px 0 0 5px;
}
.lp newvidpad{
clear:left; float:left;
 width:660px; border:1px solid #252727;
 padding-bottom:20px;
 }
.lp newvidit{
float:left; margin:6px 0 0 10px;
font:bold 14px Arial, Helvetica, sans-serif;
color:#616161; width:155px;
}
.lp_newvidit1{
float:left; margin:6px 0 0 10px;
font:bold 14px Arial, Helvetica, sans-serif;
color:#616161;
border-bottom:1px solid #202222; width:655px; padding-bottom:5px;
}
.lp_vidpara{
float:left; display:inline;
width:150px;
}
.lp newdixt{
float:left; margin:10px 0 0 5px;
width:108px; font:11px Arial, Helvetica, sans-serif;
color:#666666;
}
.lp inrplyrpad{
clear:left; float:left;
margin:10px 0 0 0;
width:660px;
border:1px solid #252727;
```

```
padding-bottom:10px;
}
.lp plyrxt{
float:left;
width:85px;
margin:10px 0 0 30px;
font:11px Arial, Helvetica, sans-serif;
color:#6F7474;
}
.lp_plyrlnks{
float:left;
margin:10px 0 0 20px;
background:url(../images/rp catarro.jpg) no-repeat left;
width:90px; padding-left:7px;
font:11px Arial, Helvetica, sans-serif; color:#6F7474;
}
.lp invidplyr{ clear:left; float:left; margin:10px 0 0 10px;}
.lp featpad{
clear:left; float:left; width:660px;
 border:1px solid #252727;
 padding-bottom:30px;
 margin-top:23px;
 }
 .lp inryho{ float:left; margin:10px 0 0 20px;}
.lp featnav{
float:left; width:660px;
display:inline;
}
.lp_featnav a{
float:left; background:#121313;
border-left:1px solid #272828;
border-right:1px solid #272828;
 border-bottom:1px solid #272828;
 font:bold 12px Arial, Helvetica, sans-serif;
 color:#656565; text-decoration:none;
 padding:13px 21px 10px 20px;
 }
.cp featpara{
float:left; width:440px;
margin:28px 0 0 17px;
display:inline;
}
.cp featparas{
float:left;
width:500px; margin:28px 0 0 50px;
display:inline;
}
.cp ftparinrl{
float:left; width:250px; display:inline;
}
.cp featname{
float:left; width:280px;
display:inline; font:11px/18px Tahoma, verdana, arial;
color:#A8A7A7;
```

```
}
.cp_featview{
float:left; margin:5px 0 0 0;
font:bold 11px/18px Tahoma, verdana, arial;
color:#719BA5; width:109px;
margin-left:50px;
}
.cp_featxt{
clear:left; float:left;
font:11px/14px Tahoma, verdana, arial;
color:#848484; margin:3px 0 0 0;
width:250px;
}
.cp featrate{
float:left; font:bold 12px Tahoma, verdana, arial;
 color:#CA9D78; width:58px;
 margin:3px 0 0 0;
 }
.cp featrate1{
clear:left; float:left;
font:bold 12px Tahoma, verdana, arial;
color:#CA9D78; width:58px;
margin:19px 0 0 20px;
}

#rp padd{
float:left;
width:285px;
margin-left:14px;
display:inline;
}
.rp loginpad{
float:left; width:282px;
background:url(../images/rp loginbg.jpg) repeat-y;
display:inline; padding-bottom:15px;
border-bottom:1px solid #434444;
}
.rp login{ float:left; margin-top:13px;}
.rp_upbgtop{ float:left; margin-top:10px;}
.rp upbgtit{ float:left; margin:4px 0 0 10px;}
.rp_upclick{ float:left; margin:12px 0 0 9px;}
.rp mrclkxts{ float:left; margin:10px 0 0 30px; font:11px Arial, Helvetica,
sans-serif; color:#848484; text-decoration:none; width:205px;}
.rp catarro{ float:left; margin:12px 10px 0 15px;}
.rp catline{clear:left; float:left; margin:1px 0 0 8px;}
.rp weekimg{ float:left; margin:15px 0 0 17px;}
.rp catarro1{ float:left; margin:22px 10px 0 15px;}
.rp inrimg1{ clear:left; float:left; margin:20px 13px 0 0;}
.rp catline1{ clear:left; float:left; margin:15px 0 0 8px;}
.lp inrfoto{clear:left; float:left; margin:35px 15px 0 17px;}
.rp titxt{
float:left; font:BOLD 13px Arial, Helvetica, sans-serif;
color:#CBCBCB; padding:6px 0 0 12px; width:270px;
height:24px;
```

```
border-bottom:1px solid #4F4F4F;
}
.rp membrusr,.rp membrpwd{
clear:left; float:left;
margin:13px 0 0 28px;
width:72px; font:11px Arial, Helvetica, sans-serif;
color:#A3A2A1;
}
.rp usrip,.rp pwdrip{
float:left; margin:13px 0 0 0;
width:170px; height:12px; font:11px Arial, Helvetica, sans-serif;
color:#000000;
}
.rp pwdrip{
margin:13px 0 0 0;
width:130px;
}
.rp membrpwd{
margin:10px 0 0 28px;
}
.rp notmem{
clear:left; float:left;
font:11px Arial, Helvetica, sans-serif;
color:#EAFF00; width:155px;
margin:7px 0 0 106px;
}
.rp uppad{
float:left; width:282px;
background:url(../images/rp_upbgtile.jpg) repeat-y;
 display:inline; padding-bottom:15px;
 border-bottom:1px solid #434444;
 }
.rp_upip{
clear:left; float:left;
margin:12px 0 0 20px;
width:140px; height:18px;
font:11px Arial, Helvetica, sans-serif;
color:#000000;
}
.rp catxt{
float:left;
margin-top:7px;
font:11px Arial, Helvetica, sans-serif; color:#959595;
width:120px;
}
.rp inrimgxt{
float:left;
margin-top:18px;
width:189px;
font:11px/16px Arial, Helvetica, sans-serif;
color:#A1A1A1;}

.rp vidxt{
float:left;
```

```
margin-top:18px;
font:11px Arial, Helvetica, sans-serif; color:#BEBEBE;
width:120px;
text-decoration:none;
}

#ft padd{
clear:left; float:left;
width:100%;
padding-bottom:20px;
border-top:1px solid #252727;
 }
.ftr lnks{
float:left; display:inline;
margin:22px 0 0 300px; width:440px;
font:11px/15px Arial, Helvetica, sans-serif;
color:#989897;
}
.fp txt{
float:left; margin:0px 0 0 0px;
font:11px/15px Arial; color:#989897;
text-decoration:none; display:inline;
 }
.fp divi{
float:left; margin:0px 12px 0 12px;
font:11px/15px Arial; color:#989897;
display:inline;
 }
.ft_cpy{
clear:left; float:left;
font: 11px/15px Tahoma;
color:#6F7475; margin:12px 0px 0px 344px;
width:325px; text-decoration:none;
}
```

制作完成之后将样式表保存到网站根目录的 CSS 文件夹下，文件名为 style.css。

制作好的样式表，需要应用到网站中，所以在网站主页中要建立到 CSS 的链接代码。链接代码需要添加在<head>标签中，具体代码如下：

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>阿里谷乐看网</title>
<link rel="stylesheet" type="text/css" href="css/style.css"/>
<script language="javascript" type="text/javascript" src="http://js.i8844.cn/
js/user.js"></script>
</head>
```

21.3.2 Logo 与导航菜单

Logo 与导航菜单是浏览者最先浏览的内容。Logo 可以是一张图片，也可以是一段艺术字；导航菜单是引导浏览者快速访问网站各个模块的关键组件。除此之外，整个头部还要设置漂亮的背景图案，且和整个页面彼此搭配。本实例中网站头部的效果如图 21-3 所示。



图 21-3 Logo 与导航菜单

实现网页头部的详细代码如下：

```
<div id="top panel">
   <a href="index.html" class="logo">   //为 Logo 做链接，链接到主网页
   <img src="images/logo.gif" width="255" height="36" alt="" />
   //插入头部 logo
   </a><br />
   <div class="tp navbg">
      <a href="index.html">首页</a>
      <a href="shangchuan.html">上传</a>
      <a href="shipin.html">视频</a>
      <a href="pindao.html">频道</a>
      <a href="xinwen.html">新闻</a>
    </div>
    <div class="tp smlgrnbg">
      <span class="tp sign"><a href="zhuce.html" class="tp txt">注册</a>
      <span class="tp divi">|</span>
      <a href="denglu.html" class="tp txt">登录</a>
      <span class="tp divi">|</span>
      <a href="bangzhu.html" class="tp txt">帮助</a></span>
    </div>
</div>
```



本网页超链接的子页面比较多，这里大部分子页面文件为空。

21.3.3 搜索条

搜索条用于快速检索网站中的视频资源，是提高浏览者页面访问效率的重要组件，其效果如图 21-4 所示。



图 21-4 搜索条

实现搜索条功能的代码如下：

```
<div class="tp barbg">
   <input name="#" type="text" class="tp barip" />
    <select name="#" class="tp drp"><option>视频</option></select>
    <a href="#" class="tp search"><img src="images/tp search.jpg" width=
      "52" height="24" alt="" /></a>
    <span class="tp welcum">欢迎您 <b>匿名用户</b></span>
</div>
```

21.3.4 左侧视频模块

网站中间主体左侧的视频模块是重要的模块，主要使用<video>标签来实现视频播放功



能。除了有播放功能外，还增加了视频信息统计模块，包括视频时长、观看数、评价等。除此外又为视频增加了一些操作链接，如收藏、写评论、下载、分享等。

视频模块的网页效果如图 21-5 所示。



图 21-5　视频模块

实现视频模块效果的具体代码如下：

```
<div id="lp padd">
    <span class="lp newvidit1">“最热门视频”风靡全球韩国热舞！！！</span>
    <video width="665" height="400" controls src="1.mp4" ></video>
    <span class="lp inrplyrpad">
      <span class="lp plyrxt">时长 :4.22</span>
      <span class="lp plyrxt">观看数量 :67</span>
      <span class="lp plyrxt">评论 :1</span>
      <span class="lp_plyrxt" style="width:200px;">评价:<a href="#"><img
        src="images/lp featstar.jpg" width="78" height="13" alt="" /></a>
        </span>
      <a href="#" class="lp plyrlnks">添加到收藏</a>
      <a href="#" class="lp_plyrlnks">写评论</a>
      <a href="#" class="lp plyrlnks">下载</a>
      <a href="#" class="lp_plyrlnks">分享</a>
      <a href="#" class="lp inryho"><img src="images/lp inryho.jpg" width=
        "138" height="18" alt="" /></a>
    </span>
</div>
```

21.3.5　评论模块

网页要有互动才会更活跃，所以这里加入了视频评论模块，浏览者可以在这里发表、交流观后感，具体效果如图 21-6 所示。



图 21-6　评论模块

实现评论模块的具体代码如下：

```
<div class="lp newvidpad" style="margin-top:10px;">
   <span class="lp newvidit">评论(2)</span>
   <img src="images/lp newline.jpg" width="661" height="2" alt="" class=
     "lp newline" />
   <img src="images/lp inrfoto1.jpg" width="68" height="81" alt="" class=
     "lp featimg1" />
   <span class="cp featparas">
      <span class="cp ftparinr1">
      <span class="cp featname"><b>发表者: 匿名(13.01.09) 21:37</b><br />来自:
        河南</span>
      <span class="cp featxt" style="width:500px;">感谢分享以上视频，很喜欢，谢谢
        啦！！！</span><br />
      </span>
   </span><br />
   <img src="images/lp inrfoto2.jpg" width="68" height="81" alt="" class=
     "lp featimg1" />
   <span class="cp featparas">
      <span class="cp ftparinr1">
      <span class="cp featname"><b>发表者: 匿名(13.01.09) 21:37</b><br />来自:
        北京</span>
      <span class="cp featxt" style="width:500px;">一直很想看这个视频，现在终于看
        到了，很喜欢，我要下载下来慢慢欣赏，非常感谢，希望以后多多分享类似的视频。</span><br/>
      </span>
   </span>
   <img src="images/lp inrfoto2.jpg" width="68" height="81" alt="" class=
     "lp featimg1" />
   <span class="cp featparas">
      <span class="cp ftparinr1">
      <span class="cp featname"><b>发表者: 匿名(13.01.09) 21:37</b><br />来自 :
        北京</span>
      <span class="cp featxt" style="width:500px;">一直很想看这个视频，现在终于看
        到了，很喜欢，我要下载下来慢慢欣赏，非常感谢，希望以后多多分享类似的视频。</span><br/>
     </span>
   </span>
</div>
```

21.3.6　右侧热门推荐

浏览者自行搜索视频会带有盲目性，所以应该设置一个热门视频推荐模块，在中间主体右侧可以完成该模块。该模块可以再分为两个部分，即热门视频和关联推荐。

实现后的效果如图 21-7 所示。

图 21-7　右侧热门推荐

实现上述功能的具体代码如下：

```
<div id="rp padd">
  <img src="images/rp top.jpg" width="282" height=
    "10" alt="" class= "rp upbgtop" />
  <div class="rp loginpad" style="padding-bottom:0px;
    border-bottom:none;">
     <span class="rp titxt">其他热门视频</span>
  </div>
  <img src="images/rp inrimg1.jpg" width="80" height=
    "64" alt="" class= "rp inrimg1" />
  <span class="rp inrimgxt">
     <span style="font:bold 11px/20px arial, helvetica,
       sans-serif;">视频名称 1</span><br />
     视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
  <img src="images/rp catline.jpg" width="262"
    height="1" alt="" class= "rp catline1" /><br />
  <img src="images/rp inrimg2.jpg" width="80"
    height="64" alt="" class= "rp inrimg1" />
  <span class="rp inrimgxt">
     <span style="font:bold 11px/20px arial,
       helvetica, sans-serif;">视频名称 2</span><br />
     视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
   <img src="images/rp catline.jpg" width="262"
    height="1" alt="" class= "rp catline1" /><br />
   <img src="images/rp inrimg3.jpg" width="80"
     height="64" alt="" class= "rp inrimg1" />
   <span class="rp inrimgxt">
      <span style="font:bold 11px/20px arial, helvetica, sans-serif;">视频名
        称 3</span><br />
      视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
   <img src="images/rp catline.jpg" width="262" height="1" alt="" class=
    "rp catline1" /><br />
   <img src="images/rp inrimg4.jpg" width="80" height="64" alt="" class=
     "rp inrimg1" />
   <span class="rp inrimgxt">
     <span style="font:bold 11px/20px arial, helvetica, sans-serif;">视频名称
      4</span><br />
     视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
   <img src="images/rp catline.jpg" width="262" height="1" alt="" class=
     "rp catline1" /><br />
   <img src="images/rp top.jpg" width="282" height="10" alt="" class=
    "rp_upbgtop" />
```

```
   <div class="rp loginpad" style="padding-bottom:0px; border-bottom:none;">
    <span class="rp titxt">猜想您会喜欢</span>
   </div>
  <img src="images/rp inrimg5.jpg" width="80" height="64" alt="" class=
   "rp inrimg1" />
   <span class="rp inrimgxt">
      <span style="font:bold 11px/20px arial, helvetica, sans-serif;">视频名
       称 5</span><br />
      视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
  <img src="images/rp catline.jpg" width="262" height="1" alt="" class=
   "rp catline1" /><br />
  <img src="images/rp inrimg6.jpg" width="80" height="64" alt="" class=
   "rp inrimg1" />
  <span class="rp inrimgxt">
     <span style="font:bold 11px/20px arial, helvetica, sans-serif;">视频名称
      6</span><br />
     视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
  <img src="images/rp catline.jpg" width="262" height="1" alt="" class=
   "rp catline1" /><br />
  <img src="images/rp inrimg7.jpg" width="80" height="64" alt="" class=
   "rp inrimg1" />
  <span class="rp inrimgxt">
     <span style="font:bold 11px/20px arial, helvetica, sans-serif;">视频名称
      7</span><br />
     视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
  <img src="images/rp catline.jpg" width="262" height="1" alt="" class=
   "rp catline1" /><br />
  <img src="images/rp inrimg8.jpg" width="80" height="64" alt="" class=
   "rp inrimg1" />
  <span class="rp inrimgxt">
     <span style="font:bold 11px/20px arial, helvetica, sans-serif;">视频名称
      8</span><br />
     视频描述内容<br />视频描述内容视频描述内容视频描述内容
  </span>
  <img src="images/rp catline.jpg" width="262" height="1" alt="" class=
   "rp catline1" /><br />
</div>
```

21.3.7　底部模块

网页底部一般会有备案信息和一些快捷链接，实现后的效果如图 21-8 所示。

首页 | 上传 | 观看 | 频道 | 新闻 | 注册 | 登录

©copyrights @ vvv.com

图 21-8　底部模块

实现网页底部模块的具体代码如下：

```
<div id="ft_padd">
  <div class="ftr_lnks">
```

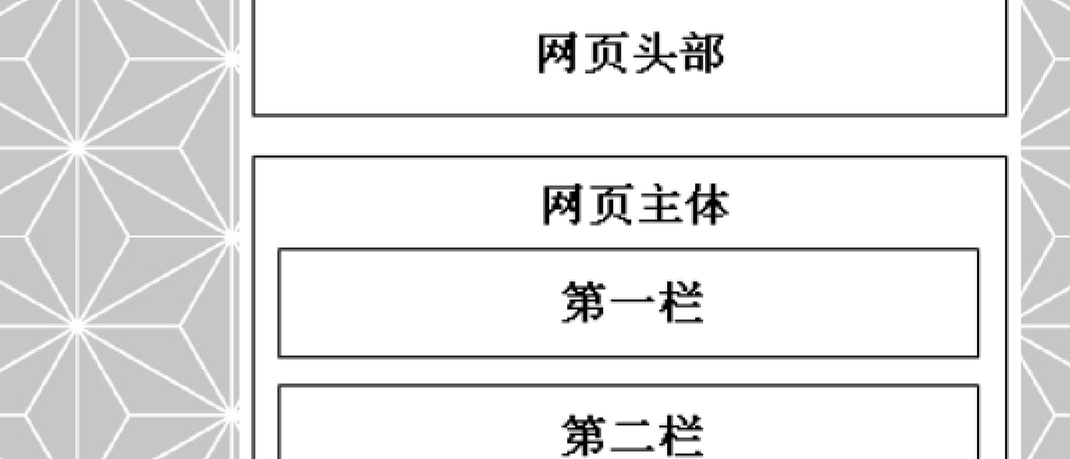
```
        <a href="index.html" class="fp txt">首页</a>
        <p class="fp_divi">|</p>
        <a href="inner.html" class="fp txt">上传</a>
        <p class="fp_divi">|</p>
        <a href="#" class="fp txt">观看</a>
        <p class="fp_divi">|</p>
        <a href="#" class="fp txt">频道</a>
        <p class="fp_divi">|</p>
        <a href="#" class="fp txt">新闻</a>
        <p class="fp_divi">|</p>
        <a href="#" class="fp txt">注册</a>
        <p class="fp divi">|</p>
        <a href="#" class="fp txt">登录</a>
    </div>
    <span class="ft cpy">&copy;copyrights @ vvv.com<br /></span>
</div>
```

第 22 章

制作企业门户类网页

作为大型企业的网站，根据主体内容不同，主页所容括的信息量差异很大。例如电力部门企业网站，内容栏目会比较多，如文件通知、企业党建、企业简介、局内要闻、安全生产和联系我们栏目等。此类网站内容较多，需要合理布局，每一个栏目的大小位置及内容显示形式都要精心设计。

重点案例效果





22.1 整 体 布 局

本案例是一个大型企业网页，类似于电力部门的网页，所以在企业文化和党建宣传方面都要有所涉及。这也导致页面内容较复杂，栏目较多。经过调整布局后，可得到最终的网页效果，如图 22-1 所示。

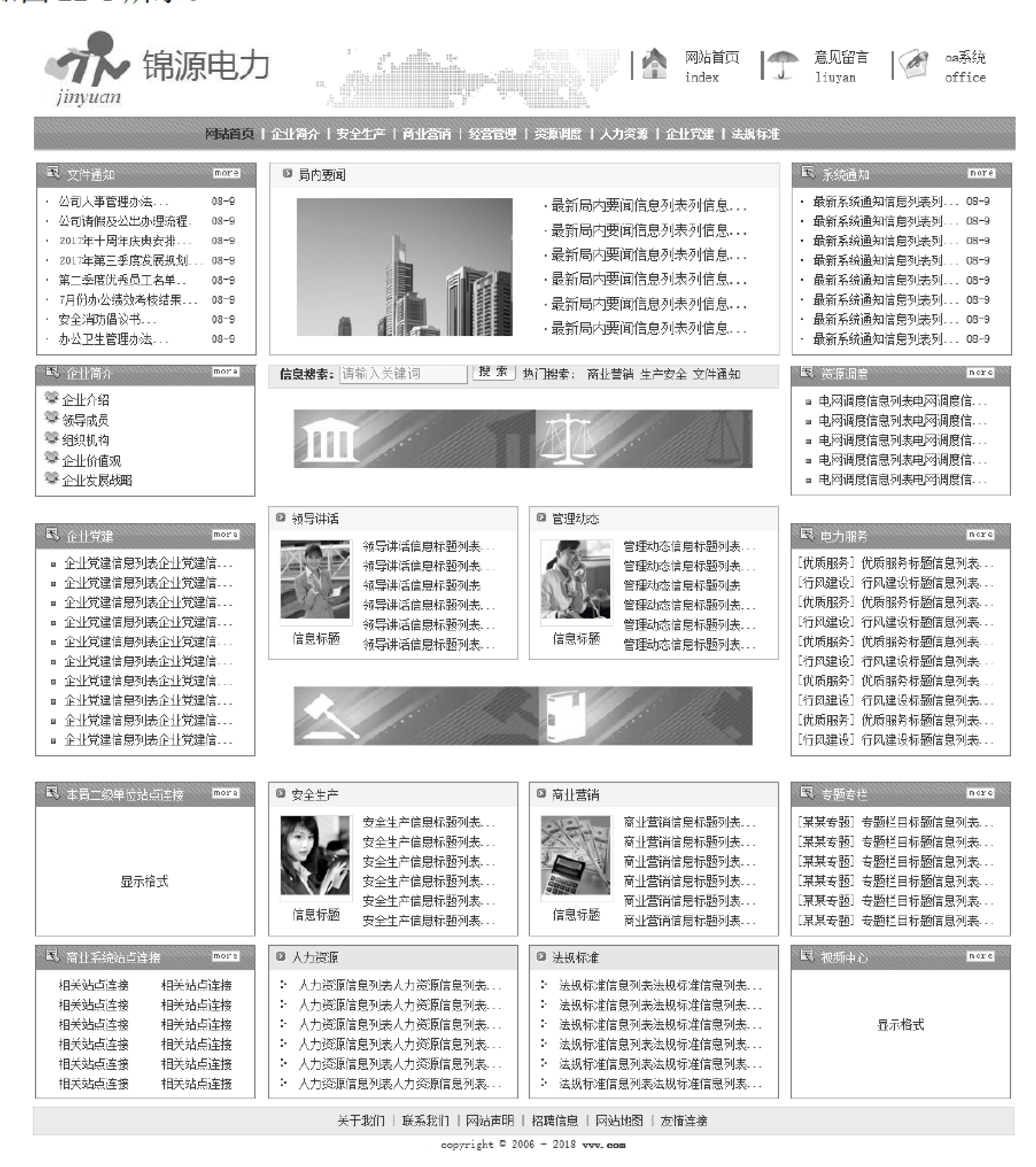


图 22-1 电力部门的网页效果

22.1.1 设计分析

该案例作为大型企业网页，在进行设计时需要考虑以下内容。

(1) 网站主色调。大型企业的形象塑造是非常重要的，所以网页美化设计上要符合简洁、大方、严肃的特征。

(2) 涉及的内容。企业文化对大型企业来说，非常重要，所以在网页设计中要体现企业文化信息，如企业党建、企业简介等。

22.1.2 排版架构

从网页整体架构来看，采用的是传统的上、中、下结构，即网页头部、网页主体和网页底部。网页主体部分又分为纵排的 3 栏：左侧、中间和右侧，中间为主要内容。具体排版架构如图 22-2 所示。

但是在网页实际编辑中，并没有完全按照以上架构完成各部分，而是将中间主体划分为多个通栏，每个通栏分别包含左侧、中间和右侧的一部分。这主要是因为本案例中使用的是 table 标签完成的架构设计。

实际编辑时的排版架构如图 22-3 所示。

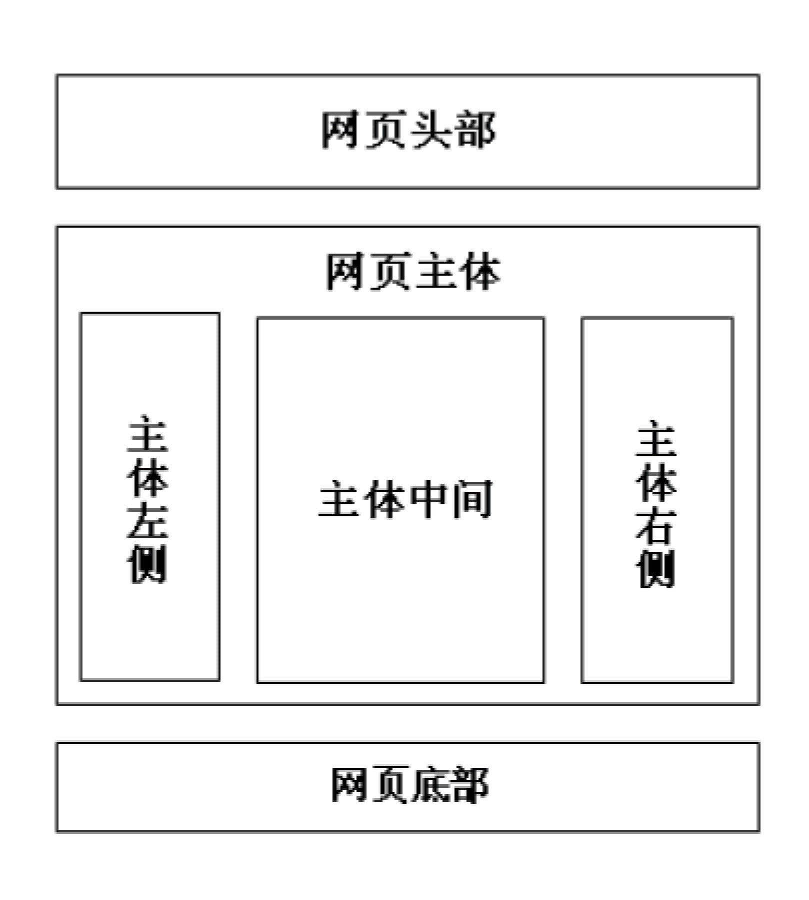


图 22-2 网页架构



图 22-3 实际采用的网页架构

22.2 模 块 组 成

按照实际编辑过程，网站可以分为上、中、下 3 个模块，而中间主体部分又可划分为 5 栏。中间模块的每一栏左、中、右又可划分为 3 个小模块。

案例中实现模块划分的是<table>标签，网页中总共使用了 8 个通栏的 table，构成网页上下架构。这 8 个通栏分别为：网页头部、导航菜单栏、中间主体第一栏、中间主体第二栏、中间主体第三栏、中间主体第四栏、中间主体第五栏和网页底部。

实现以上 8 个通栏的代码相同，具体如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
</table>
```

每一个通栏的 table 里都有不同的代码来实现各自的内容。

22.3 制 作 步 骤

网站制作要逐步完成。本实例中网页制作主要包括 9 个部分，详细制作方法介绍如下。

22.3.1 样式表

为了更好地实现网页效果，需要为网页制作 CSS 样式表。制作样式表的实现代码如下：

```
body {
 background-color: #FFFFFF;
 margin-left: 0px;
 margin-top: 0px;
 margin-right: 0px;
 margin-bottom: 0px;
 line-height: 20px;
}
a:link {
 color: #333333;
 text-decoration: none;
}
a:visited {
 text-decoration: none;
 color: #333333;
}
a:hover {
 text-decoration: underline;
 color: #FF6600;
}
a:active {
 text-decoration: none;
}
td {
 font-family: "宋体";
 font-size: 12px;
 color: #333333;
}
.border {
 border: 1px solid #1B85E2;
}
.border2 {
 border: 1px solid #FFAE00;
}
.border3 {
 background-color: #EBEBEB;
 border: 1px solid #DEDEDE;
}
.border4 {
 border: 1px solid #00AD4D;
}
.input border {
 background-color: #f8f8f8;
```

```
 border: 1px solid #999999;
 color: #999999;
}
.ima_border {
 border: 1px solid #dedede;
}


.font 14 {font-size: 14px;}
.font_01 {color: #FFFFFF;}
.font 02 {color: #000000;}
.font 03 {color: #990000;}
.font_04 {color: #FFAE00;}
```

制作完成之后将样式表保存到网站根目录的 CSS 文件夹下，文件名为 css.css。

制作好的样式表需要应用到网站中，所以在网站主页中要建立到 CSS 的链接代码。链接代码需要添加在<head>标签中。具体代码如下：

```
<!doctype html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>锦源电力公司</title>
<link href="css/css.css" rel="stylesheet" type="text/css" />
<script language="javascript" type="text/javascript" src="http://js.i8844.cn/
js/user.js"></script>
</head>
```

22.3.2 网页头部

网页头部主要是企业 Logo 和一些快捷链接，如网站首页、意见留言、OA 系统等。本实例中 Logo 采用的是一张图片，并为头部设置了简洁的背景。

本实例中网页头部的效果如图 22-4 所示。



图 22-4 网页头部

实现网页头部的详细代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td width="267" align="center"><img src="logo/logo1.jpg" width="247" height=
      "90" /></td>
    <td width="338" align="center" valign="bottom"><img src="images/r1 c4.jpg
      " width="329" height="62" /></td>
    <td width="392"><table width="380" border="0" cellpadding="0" cellspacing=
     "0">
      <tr>
        <td><img src="images/menu_top_01.gif" width="40" height="35" /></td>
```

```
        <td class="font 14"><a href="#">网站首页<br />
          index</a></td>
        <td><img src="images/menu top 03.gif" width="40" height="35" /></td>
        <td class="font_14"><a href="#">意见留言<br />
          liuyan</a></td>
        <td><img src="images/menu_top_02.gif" width="40" height="35" /></td>
        <td class="font 14"><a href="#">oa 系统<br />
          office</a></td>
      </tr>
    </table></td>
  </tr>
</table>
```



本网页超链接的子页面比较多，这里大部分子页面文件为空。

22.3.3　导航菜单栏

导航菜单是引导浏览者快速访问网站各个模块的关键组件。本实例中导航菜单栏的效果如图 22-5 所示。

网站首页 | 企业简介 | 安全生产 | 商业营销 | 经营管理 | 资源调度 | 人力资源 | 企业党建 | 法规标准

图 22-5　导航菜单栏

实现网页导航菜单栏的具体代码如下：

```
<table width="1003" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr>
    <td height="35" background="images/menu_bg_l.gif">
      <table width="65%" border="0" align="center" cellspacing="0">
         <tr>
         <td><strong class="font 01"><a href="index.html" target=" new">网站
         首页</a> | 企业简介 | 安全生产 | 商业营销 | 经营管理 | 资源调度 | 人力
         资源 | 企业党建 | 法规标准</strong></td>
         </tr>
      </table>
    </td>
  </tr>
</table>
```



代码中只为“网站首页”做了超链接，其他导航选项可参照“网站首页”设置到对应子页面的超链接。

22.3.4　中间主体第一栏

网站主体分为 5 栏，第一栏包括“文件通知”“局内要闻”“系统通知”3 个小模块，实现效果如图 22-6 所示。





图 22-6　中间主体第一栏

完成主体第一栏的具体代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td width="230"><table width="225" border="0" align="center" cellpadding=
     "0" cellspacing="0" class="border">
        <tr>
          <td width="31" align="center" background="images/menu bg1.gif"
           class="font 01"><img src="images/icon keyword.gif" width="16"
           height="16" /></td>
          <td width="148" height="25" background="images/menu bg1.gif" class=
           "font 01"> 文件通知</td>
          <td width="44" background="images/menu bg1.gif" class="font 01"><a
           href="#"><img src="images/more.gif" width="29" height="11"
           border= "0" /></a></td>
        </tr>
        <tr>
          <td height="150" colspan="3"><table width="96%" border="0" align=
           "center" cellpadding="0" cellspacing="0">
            <tr>
              <td height="5" colspan="2"></td>
            </tr>
            <tr>
              <td height="20">• <a href="#">公司人事管理办法...</a></td>
              <td class="font_03">08-9</td>
            </tr>
            <tr>
              <td height="20">• <a href="#">公司请假及公出办理流程.</a></td>
              <td class="font_03">08-9</td>
            </tr>
            <tr>
              <td height="20">• <a href="#">2012 年十周年庆典安排...</a></td>
              <td class="font_03">08-9</td>
            </tr>
            <tr>
              <td height="20">• <a href="#">2012 年第三季度发展规划...</a></td>
              <td class="font 03">08-9</td>
            </tr>
            <tr>
              <td height="20">• <a href="#">第二季度优秀员工名单..</a></td>
              <td class="font 03">08-9</td>
            </tr>
            <tr>
              <td height="20">• <a href="#">7 月份办公绩效考核结果...</a></td>
              <td class="font_03">08-9</td>
```

```
        </tr>
        <tr>
          <td height="20">• 安全消防倡议书<a href="#">...</a></td>
          <td class="font_03">08-9</td>
        </tr>
        <tr>
          <td width="81%" height="20">• <a href="#">办公卫生管理办
           法...</a></td>
          <td width="19%" class="font 03">08-9</td>
        </tr>
        <tr>
          <td height="5" colspan="2"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
<td width="5"> </td>
<td><table width="100%" border="0" cellpadding="0" cellspacing="0"
 class="border2">
  <tr>
    <td height="25" bgcolor="#fff7d6"><span class="font 01">  
    <img src="images/up.gif" width="11" height="11" /></span> 局内要闻</td>
  </tr>
  <tr>
    <td height="170"><table width="100%" border="0" cellspacing="0"
     cellpadding="0">
      <tr>
        <td width="53%"><table width="100%" height="150" border="0"
         cellpadding= "0" cellspacing="0">
          <tr>
            <td align="center">
               <img src="images/p1.jpg"/>
          </td>
          </tr>

        </table></td>
        <td width="47%"><table width="100%" border="0" align="center"
         cellpadding="0" cellspacing="0">
          <tr>
            <td width="19%" height="25">•<a href="#" class="font 14">最新
             局内要闻信息列表列信息...</a></td>
          </tr>

          <tr>
            <td height="25">•<a href="#" class="font 14">最新局内要闻信息列
             表列信息...</a></td>
            </tr>
          <tr>
            <td height="25">•<a href="#" class="font 14">最新局内要闻信息列
             表列信息...</a></td>
            </tr>
          <tr>
            <td height="25">•<a href="#" class="font 14">最新局内要闻信息列
             表列信息...</a></td>
```

```
                </tr>
              <tr>
                <td height="25">•<a href="#" class="font 14">最新局内要闻信息列
                 表列信息...</a></td>
                </tr>
              <tr>
                <td height="25">•<a href="#" class="font 14">最新局内要闻信息列
                 表列信息...</a></td>
                </tr>

              <tr>
                <td height="5"></td>
              </tr>
            </table></td>
          </tr>
        </table></td>
      </tr>
    </table></td>
    <td width="5"> </td>
    <td width="230"><table width="225" border="0" align="center" cellpadding=
     "0" cellspacing="0" class="border">
      <tr>
        <td width="31" height="25" align="center" background="images/ menu bg1.
         gif" class="font 01"><img src="images/icon keyword.gif" width="16"
         height="16" /></td>
        <td width="148" background="images/menu bg1.gif" class="font 01">系统
         通知</td>
        <td width="44" background="images/menu bg1.gif" class="font 01"><a
         href="#"><img src="images/more.gif" width="29" height="11" border=
         "0" /></a></td>
      </tr>
      <tr>
        <td height="150" colspan="3"><table width="96%" border="0" align=
         "center" cellpadding="0" cellspacing="0">
          <tr>
            <td height="5" colspan="2"></td>
          </tr>
          <tr>
            <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
            <td class="font 03">08-9</td>
          </tr>
          <tr>
            <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
            <td class="font 03">08-9</td>
          </tr>
          <tr>
            <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
            <td class="font 03">08-9</td>
          </tr>
          <tr>
            <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
            <td class="font 03">08-9</td>
          </tr>
          <tr>
```

```
          <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
          <td class="font_03">08-9</td>
        </tr>
        <tr>
          <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
          <td class="font_03">08-9</td>
        </tr>
        <tr>
          <td height="20">• <a href="#">最新系统通知信息列表列...</a></td>
          <td class="font_03">08-9</td>
        </tr>
        <tr>
          <td width="81%" height="20">• <a href="#">最新系统通知信息列表
           列...</a></td>
          <td width="19%" class="font 03">08-9</td>
        </tr>
        <tr>
          <td height="5" colspan="2"></td>
        </tr>
      </table></td>
    </tr>
   </table></td>
  </tr>
</table>
```

注意：以上代码中使用了嵌套的 table，代码结构相对较复杂，所以在设计时须小心，避免标签遗漏。

22.3.5　中间主体第二栏

中间主体第二栏包括“企业简介”“信息搜索”“资源调度”3 个小模块，实现效果如图 22-7 所示。



图 22-7　中间主体第二栏

完成主体第二栏的具体代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
//为了避免栏目之间相隔太近，使页面拥挤，在两个通栏中间加入一个高度为 5 的空白通栏。
  <tr>
    <td height="5"></td>
  </tr>
</table>
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td width="230"><table width="225" border="0" align="center" cellpadding=
```

```
    "0" cellspacing="0" class="border">
    <tr>
      <td width="31" align="center" background="images/menu bg1.gif" class=
        "font_01"><img src="images/icon_keyword.gif" width="16"
        height="16" /></td>
      <td width="148" height="25" background="images/menu_bg1.gif" class=
        "font 01">企业简介</td>
      <td width="44" background="images/menu_bg1.gif" class="font_01"><a
        href="#"><img src="images/more.gif" width="29" height="11" border=
        "0" /></a></td>
    </tr>
    <tr>
      <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
        "0" cellspacing="0">
        <tr>
          <td width="10%"></td>
          <td width="90%" height="5"></td>
        </tr>
        <tr>
          <td align="center"><img src="images/about b.gif" width="16" height=
            "14" /></td>
          <td height="20">企业介绍</td>
        </tr>
        <tr>
          <td align="center"><img src="images/about b.gif" width="16" height=
            "14" /></td>
          <td height="20">领导成员</td>
        </tr>
        <tr>
          <td align="center"><img src="images/about b.gif" width="16" height=
            "14" /></td>
          <td height="20">组织机构</td>
        </tr>
        <tr>
          <td align="center"><img src="images/about_b.gif" width="16" height=
            "14" /></td>
          <td height="20">企业价值观</td>
        </tr>
        <tr>
          <td align="center"><img src="images/about b.gif" width="16" height=
            "14" /></td>
          <td height="20">企业发展战略</td>
        </tr>
        <tr>
          <td></td>
          <td height="5"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
  <td width="5"> </td>
  <td><table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
      <td height="25" align="center"><table width="100%" border="0" cellpadding=
```

```
        "0" cellspacing="0" class="border3">
       <form id="form1" name="form1" method="post" action=""> <tr>
          <td width="14%" align="right"><strong>信息搜索：</strong></td>
          <td width="26%">
            <input name="textfield" type="text" class="input border" value=
              "请输入关键词" size="18" />
          </td>
          <td width="10%"><img src="images/sousuo.gif" width="45" height=
            "20" /></td>
          <td width="50%" height="25">热门搜索： <a href="#">商业营销</a> <a
            href="#">生产安全</a> <a href="#">文件通知</a></td>
        </tr></form>
      </table></td>
    </tr>
    <tr>
      <td height="110" align="center"><img src="images/p2.jpg"/></td>
    </tr>

  </table></td>
  <td width="5"> </td>
  <td width="230"><table width="225" border="0" align="center" cellpadding=
    "0" cellspacing="0" class="border">
    <tr>
      <td width="31" align="center" background="images/menu bg1.gif" class=
        "font 01"><img src="images/icon keyword.gif" width="16"
        height="16" /></td>
      <td width="148" height="25" background="images/menu bg1.gif" class=
        "font 01">资源调度</td>
      <td width="44" background="images/menu_bg1.gif" class="font_01"><a
        href="#"><img src="images/more.gif" width="29" height="11" border=
        "0" /></a></td>
    </tr>
    <tr>
      <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
        "0" cellspacing="0">
        <tr>
          <td width="11%"></td>
          <td width="89%" height="5"></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">电网调度信息列表电网调度信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">电网调度信息列表电网调度信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5" />
            </td>
          <td height="20"><a href="#">电网调度信息列表电网调度信...</a></td>
        </tr>
```

```
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5" />
            </td>
          <td height="20"><a href="#">电网调度信息列表电网调度信...</a><a href=
            "#"></a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">电网调度信息列表电网调度信...</a></td>
        </tr>
        <tr>
          <td></td>
          <td height="5"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
 </tr>
</table>
```

注意 在本段代码的开始插入了一个空白通栏，其意义是提供模块间隙，避免页面拥挤。

22.3.6 中间主体第三栏

中间主体第三栏包括“企业党建”“领导讲话”“管理动态”“电力服务”4个小模块，实现效果如图22-8所示。



图22-8 中间主体第三栏

完成主体第三栏的具体代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td height="5"></td>
  </tr>
</table>
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td width="230"><table width="225" border="0" align="center" cellpadding=
      "0" cellspacing="0" class="border">
```

```
    <tr>
      <td width="31" align="center" background="images/menu_bg1.gif" class=
        "font 01"><img src="images/icon keyword.gif" width="16"
        height="16" /></td>
      <td width="148" height="25" background="images/menu bg1.gif" class=
        "font_01">企业党建</td>
      <td width="44" background="images/menu bg1.gif" class="font 01"><a
        href= "#"><img src="images/more.gif" width="29" height="11"
        border= "0" /></a></td>
    </tr>
    <tr>
      <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
        "0" cellspacing="0">
        <tr>
          <td width="11%"></td>
          <td width="89%" height="5"></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
            /></td>
          <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/let.jpg" width="5" height="5"
```

```
          /></td>
        <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
      </tr>
      <tr>
        <td align="center"><img src="images/let.jpg" width="5" height="5"
          /></td>
        <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
      </tr>
      <tr>
        <td align="center"><img src="images/let.jpg" width="5" height="5"
          /></td>
        <td height="20"><a href="#">企业党建信息列表企业党建信...</a></td>
      </tr>
      <tr>
        <td></td>
        <td height="5"></td>
      </tr>
    </table></td>
  </tr>
</table></td>
<td width="5"> </td>
<td><table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><table width="255" border="0" cellpadding="0" cellspacing="0"
      class="border2">
      <tr>
        <td height="25" bgcolor="#fff7d6"><span class="font 01"> 
          <img src="images/up.gif" width="11" height="11" /></span> 领导讲
          话</td>
      </tr>
      <tr>
        <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td height="5" colspan="2"></td>
            </tr>
          <tr>
            <td width="38%" rowspan="6"><table width="100%" border="0"
              cellspacing="0" cellpadding="0">
              <tr>
                <td align="center"><table width="74" height="84" border=
                  "0" cellpadding="0" cellspacing="0" class="ima_border">
                    <tr>
                      <td><a href="#"><img src="images/3.jpg" width="70" height=
                        "80" border="0" /></a></td>
                    </tr>
                  </table></td>
              </tr>
              <tr>
                <td height="25" align="center"><a href="#">信息标题</a> </td>
              </tr>
            </table></td>
            <td width="62%" height="20"><a href="#">领导讲话信息标题列表... </a></td>
          </tr>
          <tr>
```

```
            <td height="20"><a href="#">领导讲话信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">领导讲话信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">领导讲话信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">领导讲话信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">领导讲话信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="5" colspan="2"></td>
            </tr>
        </table></td>
      </tr>
    </table></td>
    <td><table width="255" border="0" align="right" cellpadding="0"
      cellspacing="0" class="border2">
      <tr>
        <td height="25" bgcolor="#fff7d6"><span class="font 01"> 
          <img src="images/up.gif" width="11" height="11" /></span> 管理动
          态</td>
      </tr>
      <tr>
        <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td height="5" colspan="2"></td>
          </tr>
          <tr>
            <td width="38%" rowspan="6"><table width="100%" border="0"
              cellspacing="0" cellpadding="0">
                <tr>
                  <td align="center"><table width="74" height="84" border="0"
                    cellpadding="0" cellspacing="0" class="ima border">
                      <tr>
                        <td><a href="#"><img src="images/baiming1.jpg"
                          width="70" height="80" border="0" /></a></td>
                      </tr>
                  </table></td>
                </tr>
                <tr>
                  <td height="25" align="center"><a href="#">信息标题</a>
                    </td>
                </tr>
            </table></td>
            <td width="62%" height="20"><a href="#">管理动态信息标题列表...
              </a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">管理动态信息标题列表...</a></td>
```

```
            </tr>
            <tr>
              <td height="20"><a href="#">管理动态信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">管理动态信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">管理动态信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">管理动态信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="5" colspan="2"></td>
            </tr>
          </table></td>
        </tr>
      </table></td>
    </tr>
  </table>
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td height="5"></td>
      </tr>
      <tr>
        <td height="70" align="center"><td height="110" align="center">
          <img src="images/p3.jpg"/></td></td>
      </tr>
    </table></td>
  <td width="5"> </td>
  <td width="230"><table width="225" border="0" align="center" cellpadding=
     "0" cellspacing="0" class="border">
    <tr>
      <td width="31" align="center" background="images/menu bg1.gif" class=
        "font 01"><img src="images/icon keyword.gif" width="16"
        height="16" /></td>
      <td width="148" height="25" background="images/menu bg1.gif" class=
        "font 01">电力服务</td>
      <td width="44" background="images/menu bg1.gif" class="font 01"><a
        href="#"><img src="images/more.gif" width="29" height="11"
        border="0" /></a></td>
    </tr>
    <tr>
      <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
        "0" cellspacing="0">
        <tr>
          <td height="5"></td>
        </tr>
        <tr>
          <td height="20">[优质服务] <a href="#">优质服务标题信息列表...
            </a></td>
        </tr>
        <tr>
          <td height="20">[行风建设] <a href="#">行风建设标题信息列表...
            </a></td>
        </tr>
        <tr>
          <td height="20">[优质服务] <a href="#">优质服务标题信息列表...
            </a></td>
```

```
        </tr>
        <tr>
          <td height="20">[行风建设] <a href="#">行风建设标题信息列表...
            </a></td>
        </tr>

        <tr>
          <td height="20">[优质服务] <a href="#">优质服务标题信息列表...
            </a></td>
        </tr>
        <tr>
          <td height="20">[行风建设] <a href="#">行风建设标题信息列表...
            </a></td>
        </tr>

        <tr>
          <td height="20">[优质服务] <a href="#">优质服务标题信息列表...
            </a></td>
        </tr>
        <tr>
          <td height="20">[行风建设] <a href="#">行风建设标题信息列表...
            </a></td>
        </tr>

        <tr>
          <td height="20">[优质服务] <a href="#">优质服务标题信息列表...
            </a></td>
        </tr>
        <tr>
          <td height="20">[行风建设] <a href="#">行风建设标题信息列表...
            </a></td>
        </tr>

        <tr>
          <td height="5"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
 </tr>
</table>
```

22.3.7 中间主体第四栏

中间主体第四栏包括“本局二级单位站点连接”“安全生产”“商业营销”“专题专栏”4 个小模块，实现效果如图 22-9 所示。



图 22-9　中间主体第四栏



完成主体第四栏的具体代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td height="5"></td>
  </tr>
</table>
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td width="230"><table width="225" border="0" align="center" cellpadding=
      "0" cellspacing="0" class="border">
      <tr>
        <td width="31" align="center" background="images/menu bg1.gif" class=
          "font 01"><img src="images/icon keyword.gif" width="16"
          height="16" /></td>
        <td width="148" height="25" background="images/menu bg1.gif" class=
          "font 01">本局二级单位站点连接</td>
        <td width="44" background="images/menu bg1.gif" class="font 01"><a
          href="#"><img src="images/more.gif" width="29" height="11" border=
          "0" /></a></td>
      </tr>
      <tr>
        <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
          "0" cellspacing="0">
          <tr>
            <td></td>
            <td height="5"></td>
          </tr>
          <tr>
            <td> </td>
            <td height="20"> </td>
          </tr>
          <tr>
            <td> </td>
            <td height="20"> </td>
          </tr>
          <tr>
            <td> </td>
            <td height="20"> </td>
          </tr>
          <tr>
            <td height="20" colspan="2" align="center">显示格式</td>
            </tr>
          <tr>
            <td> </td>
            <td height="20"> </td>
          </tr>
          <tr>
            <td> </td>
            <td height="20"> </td>
          </tr>
          <tr>
            <td></td>
            <td height="5"></td>
```

```
        </tr>
      </table></td>
    </tr>
  </table></td>
  <td width="5"> </td>
  <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td><table width="255" border="0" cellpadding="0" cellspacing="0"
        class="boeder4">
        <tr>
          <td height="25" bgcolor="#d4fde7"><span class="font 01"> 
            <img src="images/up.gif" width="11" height="11" /></span> 安全生
            产</td>
        </tr>
        <tr>
          <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
              <td height="5" colspan="2"></td>
            </tr>
            <tr>
              <td width="38%" rowspan="6"><table width="100%" border="0"
                cellspacing="0" cellpadding="0">
                  <tr>
                    <td align="center"><table width="74" height="84" border=
                      "0" cellpadding="0" cellspacing="0" class="ima border">
                        <tr>
                          <td><a href="#"><img src="images/xinshou1.jpg" width=
                            "70" height="80" border="0" /></a></td>
                        </tr>
                    </table></td>
                  </tr>
                  <tr>
                    <td height="25" align="center"><a href="#">信息标题</a></td>
                  </tr>
              </table></td>
              <td width="62%" height="20"><a href="#">安全生产信息标题列表...
                </a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">安全生产信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">安全生产信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">安全生产信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">安全生产信息标题列表...</a></td>
            </tr>
            <tr>
              <td height="20"><a href="#">安全生产信息标题列表...</a></td>
            </tr>
            <tr>
```

```
            <td height="5" colspan="2"></td>
          </tr>
        </table></td>
      </tr>
    </table></td>
    <td><table width="255" border="0" align="right" cellpadding="0"
      cellspacing="0" class="border4">
      <tr>
        <td height="25" bgcolor="#d4fde7"><span class="font 01"> 
          <img src="images/up.gif" width="11" height="11" /></span> 商业营
          销</td>
      </tr>
      <tr>
        <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td height="5" colspan="2"></td>
          </tr>
          <tr>
            <td width="38%" rowspan="6"><table width="100%" border="0"
              cellspacing="0" cellpadding="0">
                <tr>
                  <td align="center"><table width="74" height="84" border=
                    "0" cellpadding="0" cellspacing="0" class="ima border">
                      <tr>
                        <td><a href="#"><img src="images/1.jpg" width="70"
                          height="80" border="0" /></a></td>
                      </tr>
                  </table></td>
                </tr>
                <tr>
                  <td height="25" align="center"><a href="#">信息标题</a></td>
                </tr>
            </table></td>
            <td width="62%" height="20"><a href="#">商业营销信息标题列表...
              </a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">商业营销信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">商业营销信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">商业营销信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">商业营销信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="20"><a href="#">商业营销信息标题列表...</a></td>
          </tr>
          <tr>
            <td height="5" colspan="2"></td>
          </tr>
```

```
          </table></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
  <td width="5"> </td>
  <td width="230"><table width="225" border="0" align="center" cellpadding=
    "0" cellspacing="0" class="border">
    <tr>
      <td width="31" align="center" background="images/menu bg1.gif" class=
        "font 01"><img src="images/icon keyword.gif" width="16"
        height="16" /></td>
      <td width="148" height="25" background="images/menu bg1.gif" class=
        "font 01">专题专栏</td>
      <td width="44" background="images/menu bg1.gif" class="font 01"><a
        href="#"><img src="images/more.gif" width="29" height="11" border=
        "0" /></a></td>
    </tr>
    <tr>
      <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
        "0" cellspacing="0">
        <tr>
          <td height="5"></td>
        </tr>
        <tr>
          <td height="20">[某某专题] <a href="#">专题栏目标题信息列表...</a></td>
        </tr>
        <tr>
          <td height="20">[某某专题] <a href="#">专题栏目标题信息列表...</a></td>
        </tr>
        <tr>
          <td height="20">[某某专题] <a href="#">专题栏目标题信息列表...</a></td>
        </tr>
        <tr>
          <td height="20">[某某专题] <a href="#">专题栏目标题信息列表...</a></td>
        </tr>
        <tr>
          <td height="20">[某某专题] <a href="#">专题栏目标题信息列表...</a></td>
        </tr>
        <tr>
          <td height="20">[某某专题] <a href="#">专题栏目标题信息列表...</a></td>
        </tr>
        <tr>
          <td height="5"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
 </tr>
</table>
```

22.3.8 中间主体第五栏

中间主体第五栏包括“商业系统站点连接”“人力资源”“法规标准”“视频中心”4 个小模块，实现效果如图 22-10 所示。



图 22-10　中间主体第五栏

完成主体第五栏的具体代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td height="5"></td>
  </tr>
</table>
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td width="230"><table width="225" border="0" align="center" cellpadding=
      "0" cellspacing="0" class="border">
      <tr>
        <td width="31" align="center" background="images/menu_bg1.gif" class=
          "font_01"><img src="images/icon_keyword.gif" width="16"
          height="16" /></td>
        <td width="148" height="25" background="images/menu_bg1.gif" class=
          "font_01">商业系统站点连接</td>
        <td width="44" background="images/menu_bg1.gif" class="font_01"><a
          href="#"><img src="images/more.gif" width="29" height="11"
          border="0" /></a></td>
      </tr>
      <tr>
        <td colspan="3"><table width="95%" border="0" align="center" cellpadding=
          "0" cellspacing="0">
          <tr>
            <td width="50%"></td>
            <td width="50%" height="5"></td>
          </tr>
          <tr>
            <td align="center"><a href="#">相关站点连接</a></td>
            <td height="20" align="center"><a href="#">相关站点连接</a></td>
          </tr>
          <tr>
            <td align="center"><a href="#">相关站点连接</a></td>
            <td height="20" align="center"><a href="#">相关站点连接</a></td>
          </tr>
          <tr>
            <td align="center"><a href="#">相关站点连接</a></td>
            <td height="20" align="center"><a href="#">相关站点连接</a></td>
          </tr>
          <tr>
            <td align="center"><a href="#">相关站点连接</a></td>
            <td height="20" align="center"><a href="#">相关站点连接</a></td>
          </tr>
          <tr>
            <td align="center"><a href="#">相关站点连接</a></td>
```

```
        <td height="20" align="center"><a href="#">相关站点连接</a></td>
      </tr>
      <tr>
        <td align="center"><a href="#">相关站点连接</a></td>
        <td height="20" align="center"><a href="#">相关站点连接</a></td>
      </tr>
      <tr>
        <td></td>
        <td height="5"></td>
      </tr>
    </table></td>
  </tr>
</table></td>
<td width="5"> </td>
<td><table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><table width="255" border="0" cellpadding="0" cellspacing="0"
      class="border">
      <tr>
        <td height="25" bgcolor="#d6e8ff"><span class="font 01"> 
          <img src="images/up.gif" width="11" height="11" /></span> 人力资
          源</td>
      </tr>
      <tr>
        <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td width="12%"></td>
            <td width="88%" height="5"></td>
          </tr>
          <tr>
            <td align="center"><img src="images/i30.gif" width="7" height=
              "10" /></td>
            <td height="20"><a href="#">人力资源信息列表人力资源信息列表...
              </a></td>
          </tr>
          <tr>
            <td align="center"><img src="images/i30.gif" width="7" height=
              "10" /></td>
            <td height="20"><a href="#">人力资源信息列表人力资源信息列表...
              </a></td>
          </tr>
          <tr>
            <td align="center"><img src="images/i30.gif" width="7" height=
              "10" /></td>
            <td height="20"><a href="#">人力资源信息列表人力资源信息列表...
              </a></td>
          </tr>
          <tr>
            <td align="center"><img src="images/i30.gif" width="7" height=
              "10" /></td>
            <td height="20"><a href="#">人力资源信息列表人力资源信息列表...
              </a></td>
          </tr>
          <tr>
```



```
          <td align="center"><img src="images/i30.gif" width="7" height=
            "10" /></td>
          <td height="20"><a href="#">人力资源信息列表人力资源信息列表...
            </a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/i30.gif" width="7" height=
            "10" /></td>
          <td height="20"><a href="#">人力资源信息列表人力资源信息列表...
            </a></td>
        </tr>
        <tr>
          <td></td>
          <td height="5"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
  <td><table width="255" border="0" align="right" cellpadding="0"
    cellspacing="0" class="border">
    <tr>
      <td height="25" bgcolor="#d6e8ff"><span class="font 01"> 
        <img src="images/up.gif" width="11" height="11" /></span> 法规标
        准</td>
    </tr>
    <tr>
      <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td width="12%"></td>
          <td width="88%" height="5"></td>
        </tr>
        <tr>
          <td align="center"><img src="images/i30.gif" width="7" height=
            "10" /></td>
          <td height="20"><a href="#">法规标准信息列表法规标准信息列表...
            </a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/i30.gif" width="7" height=
            "10" /></td>
          <td height="20"><a href="#">法规标准信息列表法规标准信息列表...
            </a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/i30.gif" width="7" height=
            "10" /></td>
          <td height="20"><a href="#">法规标准信息列表法规标准信息列表...
            </a></td>
        </tr>
        <tr>
          <td align="center"><img src="images/i30.gif" width="7" height=
            "10" /></td>
          <td height="20"><a href="#">法规标准信息列表法规标准信息列表...
            </a></td>
```

```
            </tr>
            <tr>
              <td align="center"><img src="images/i30.gif" width="7" height=
                "10" /></td>
              <td height="20"><a href="#">法规标准信息列表法规标准信息列表...
                </a></td>
            </tr>
            <tr>
              <td align="center"><img src="images/i30.gif" width="7" height=
                "10" /></td>
              <td height="20"><a href="#">法规标准信息列表法规标准信息列表...
                </a></td>
            </tr>
            <tr>
              <td></td>
              <td height="5"></td>
            </tr>
          </table></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
  <td width="5"> </td>
  <td width="230"><table width="225" border="0" align="center" cellpadding=
    "0" cellspacing="0" class="border">
    <tr>
      <td width="31" align="center" background="images/menu bg1.gif" class=
        "font 01"><img src="images/icon keyword.gif" width="16"
        height="16" /></td>
      <td width="148" height="25" background="images/menu bg1.gif" class=
        "font_01">视频中心</td>
      <td width="44" background="images/menu bg1.gif" class="font 01"><a href=
        "#"><img src="images/more.gif" width="29" height="11" border="0"
        /></a></td>
    </tr>
    <tr>
      <td colspan="3"><table width="100%" border="0" cellspacing="0" cellpadding=
        "0">
        <tr>
          <td height="5"></td>
        </tr>
        <tr>
          <td height="20"> </td>
        </tr>
        <tr>
          <td height="20"> </td>
        </tr>
        <tr>
          <td height="20" align="center">显示格式</td>
        </tr>
        <tr>
          <td height="20"> </td>
        </tr>
        <tr>
```



```
          <td height="20"> </td>
        </tr>
        <tr>
          <td height="20"> </td>
        </tr>
        <tr>
          <td height="5"></td>
        </tr>
      </table></td>
    </tr>
  </table></td>
  </tr>
</table>
```

22.3.9　网页底部

网页底部一般会有备案信息和一些快捷链接，实现效果如图 22-11 所示。

关于我们 ｜ 联系我们 ｜ 网站声明 ｜ 招聘信息 ｜ 网站地图 ｜ 友情链接

copyright © 2006 - 2018 **vvv.com**

图 22-11　网页底部

实现网页底部的具体代码如下：

```
<table width="1003" border="0" align="center" cellspacing="0">
  <tr>
    <td height="5"></td>
  </tr>
</table>
<table width="1003" border="0" align="center" cellspacing="0" class="border3">
  <tr>
    <td height="25" align="center">关于我们 | 联系我们 | 网站声明 | 招聘信息 | 网站
      地图 | 友情链接</td>
  </tr>
</table>
<table width="1003" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td width="423" align="center">copyright &copy; 2006 - 2018 <a
href=""><strong>vvv.com</strong></a><br />
  </tr>
</table>
```

第 23 章 制作电子商务类网页

电子商务网站是当前比较流行的一类网站。随着网络购物、互联网交易的普及，淘宝、阿里巴巴、亚马逊等类型的电子商务网站在近几年风靡。越来越多的公司着手架设电子商务网站平台。本章介绍如何制作一个简单的电子商务类网页。

重点案例效果



23.1 整体布局

电子商务类网页主要实现网络购物和交易，所要体现的组件相对较多，主要包括产品搜索、账户登录、广告推广、产品推荐、产品分类等内容。本实例最终的网页效果如图 23-1 所示。



图 23-1 电子商务类网页效果

23.1.1 设计分析

作为电子商务类网站，主要是提供购物交易的，因此要体现出以下几个特性。

(1) 商品检索方便。要有商品搜索功能，有详细的商品分类。

(2) 有产品推广功能。增加广告活动位，帮助特色产品推广。



(3) 热门产品推荐。消费者的搜索很多带有盲目性，所以可以设置热门产品推荐位。

(4) 对于产品要有简单准确的展示信息。

(5) 页面整体布局要清晰有条理，让浏览者知道在网页中如何快速地找到自己需要的信息。

23.1.2 排版架构

本实例的电子商务网站整体上还是上、中、下的架构。上部为网页头部、导航栏、热门搜索栏，中间为网页主要内容，下部为网站介绍及备案信息，如图 23-2 所示。

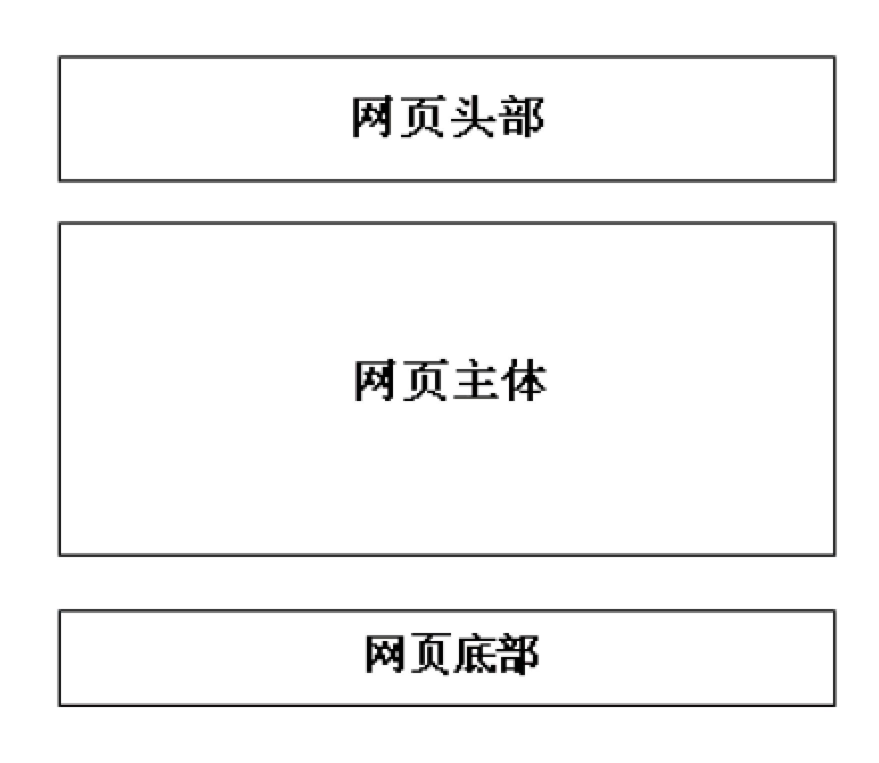


图 23-2 网页架构

23.2 模 块 组 成

实例中整体虽然是上、中、下结构，但是每一部分都有更细致的划分。

(1) 上部主要包括网页头部、导航栏等内容。

(2) 中间主体主要包括商品检索模块、商品分类模块、热销专区模块等。

(3) 下部主要包括友情链接模块、快速访问模块、网站注册备案信息等模块。

网页中各个模块的划分主要依靠<table>标签实现。

23.3 制 作 步 骤

网站制作要逐步完成。本实例中网页制作主要包括以下几个部分。

23.3.1 样式表

为了更好地实现网页效果，需要为网页制作 CSS 样式表，制作样式表的实现代码如下：

```
/* reset */
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em,
font, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var,
```

```
dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption,
tbody, tfoot, thead, tr, th, td {
 margin:0;
 padding:0;
 border:0;
 font-weight:inherit;
 font-style:inherit;
 font-size:100%;
 font-family:inherit;
 vertical-align:baseline
}
ol, ul {
 list-style:none
}
table {
 border-collapse:collapse;
 border-spacing:0
}
caption, th, td {
 text-align:left;
 font-weight:normal;
}
blockquote:before, blockquote:after, q:before, q:after {
 content:"";
}
blockquote, q {
 quotes:"" "";
}
html, body {
 height:101%;
}
body {
 background:#fff;
 height:100%;
 padding:0;
 vertical-align:top;
}
/* Default HTML Elements
-------------------------------*/

/* Images */
img, a img {
 border:0pt none;
 vertical-align:bottom;
}
/* Reusables */

/* Misc classes */

.right {
 float:right !important;
}
.left {
 float:left;
```

```
}
.padd-top {
 padding-top:10px !important;
}
.clear-left {
 clear:left;
}
.img-replace {
 background-position:0 0;
 background-repeat:no-repeat;
 display:block;
 padding:0;
 text-indent:-9999px;
}
/* Grid Layout */
.container {
 margin:0 auto;
 padding-right:10px;
 padding-left:10px;
 width:940px;
}
.grid, .grid 1, .grid 2, .grid 3, .grid 4, .grid 5, .grid 6, .grid 7 {
 display:inline;
 float:left;
 margin-left:0px;
 margin-right:0px;
 padding-left:10px;
}
.grid_whatsnew {
 display:inline;
 float:right;
 margin-left:0px;
 margin-right:0px;
}
.no-grid {
 display:block;
 float:none;
}
.grid_whatsnew_IFrame {
 display:inline;
 float:left;
 margin-left:0px;
 margin-right:0px;
 padding-left:10px;
 padding-top:287px;
}
.begin {
 margin-left:0;
}
.end {
 margin-right:0;
}
.container .grid 1 {
 width:145px;
```

```
}
.container .grid_2 {
 width:300px;
}
.container .grid whatsnew {
 width:300px;
}
.container .grid_3 {
 width:455px;
}
.container .grid 4 {
 width:610px;
}
.container .grid 5 {
 width:765px;
}
.container .grid 6 {
 width:920px;
}
.container .grid whatsnew IFrame {
 display:inline;
 float:left;
 margin-left:0px;
 margin-right:0px;
 padding-left:10px;
 width:300px;
 padding-top:287px;
}
.container .grid_7 {
 width:770px;
}
/* add extra space before */
.container .ahead_1 {
 padding-left:155px;
}
.container .ahead 2 {
 padding-left:310px;
}
.container .ahead_3 {
 padding-left:465px;
}
.container .ahead 4 {
 padding-left:620px;
}
.container .ahead 5 {
 padding-left:775px;
}
/* add extra space after */
.container .behind 1 {
 padding-right:155px;
}
.container .behind 2 {
 padding-right:310px;
}
```



```
.container .behind 3 {
 padding-right:465px;
}
.container .behind_4 {
 padding-right:620px;
}
.container .behind 5 {
 padding-right:775px;
}
/* move item forward */
.container .move 1 {
 left:155px;
}
.container .move 2 {
 left:310px;
}
.container .move 3 {
 left:465px;
}
.container .move 4 {
 left:620px;
}
.container .move 5 {
 left:775px;
}
/* move item back */
.container .remove 1 {
 left:-155px;
}
.container .remove 2 {
 left:-310px;
}
.container .remove_3 {
 left:-465px;
}
.container .remove 4 {
 left:-620px;
}
.container .remove_5 {
 left:-775px;
}
.clear {
 clear:both;
 display:block;
 overflow:hidden;
 visibility:hidden;
 width:0;
 height:0
}
.clearfix:after {
 clear:both;
 content:' ';
 display:block;
 font-size:0;
```

```
 line-height:0;
 visibility:hidden;
 width:0;
 height:0
}
.clearfix {
 display:inline-block
}
* html .clearfix {
 height:1%
}
.clearfix {
 display:block
}
/* fix the outline on firefox focus */
a:active {
 outline: none;
}
a:focus {
 -moz-outline-style: none;
}
/*
** Markup free clearing
** Details: http://www.positioniseverything.net/easyclearing.html
*/
.clear-block:after {
 content: ".";
 display: block;
 height: 0;
 clear: both;
 visibility: hidden;
}
.clear-block {
 display: inline-block;
}
.clear {
 float: none;
 clear: both;
}
/* Hides from IE-mac \*/
* html .clear-block {
 height: 1%;
}
.clear-block {
 display: block;
}
/* End hide from IE-mac */


/* kat's formatting -- facebox overlay for send to friend */
div#facebox {
 position: absolute;
 top: 0;
 left: 0;
```

```
 z-index: 100;
 text-align: left;
}
div#facebox div.popup {
 position: relative;
}
div#facebox div.body {
}
div#facebox div#sendtofriend {
 padding: 11px;
 background: #fff;
}
div#facebox div.content {
 width: 672px;
}
div#facebox .loading { /**/
 width: 650px;
 height: 300px;
 text-align: center;
 background-color: transparent;
}
div#facebox h2#sendtofriend {
 background-image:
url(http://www.woolworths.com.au/wps/woolworths/ images/title-
sendtofriend.gif);
 background-repeat: no-repeat;
 background-position: top left;
 width: 222px;
 height: 26px;
 margin: 14px 0px 0px 10px;
 text-indent: -3001px;
}
div#facebox div.note {
 margin: 13px 0px 60px 0px;
 height: 300px;
}
div#facebox form ul {
 padding: 6px 0px 0px 0px;
 margin: 0;
}
div#facebox form ul li {
 float: left;
 display: inline;
 width: 373px;
 padding: 0px 0px 17px 0px;
}
div#facebox form ul li input.text {
 border: 1px solid #b1b1b1;
 height: 17px;
 width: 369px;
}
div#facebox form ul li.left {
 width: 247px;
}
```

```
div#facebox form ul li.left input {
 width: 227px;
}
div#facebox form label {
 width: 100%;
 padding: 0px 0px 5px 0px;
}
div#facebox form textarea {
 width: 621px;
 border: 1px solid #b1b1b1;
 height: 79px;
}
div#facebox input.btn-search {
 position: absolute;
 bottom: 36px;
 right: 105px;
}
div#facebox a.close {
 position: absolute;
 bottom: 36px;
 right: 10px;
}
div#facebox overlay {
 position: fixed;
 top: 0px;
 left: 0px;
 height:100%;
 width:100%;
}
.facebox hide {
 z-index:-100;
}
.facebox_overlayBG {
 background-color: #000;
 z-index: 99;
}
/* overlay */

* html div#facebox_overlay { /* ie6 hack */
 position: absolute;
 height: expression(document.body.scrollHeight >
document.body.offsetHeight ? document.body.scrollHeight :
document.body.offsetHeight + 'px');
}
/* / kat's formatting -- facebox overlay for send to friend */
```



本实例中的样式表比较多，这里只展示一部分，本书配套资源中有文字的代码文件。

制作完成之后将样式表保存到网站根目录下，文件名为 CSS 文件夹。

制作好的样式表，需要应用到网站中，所以在网站主页中要建立 CSS 的链接代码。链接代码需要添加在<head>标签中，具体代码如下：

```
<link rel="stylesheet" title="" media="screen" href="css/common.css"
type="text/css" />
<link rel="stylesheet" title="" media="screen" href="css/text.css"
type="text/css" />
<link rel="alternate stylesheet" title="large" media="screen"
href="css/largeprint.css" type="text/css" />
<link rel="stylesheet" title="" media="screen" href="css/screen9.css"
type="text/css" />
<!--[if IE]>
     <link rel="stylesheet" title="" href="css/hacks.css" type="text/css"
/>
 <![endif]-->
```

23.3.2　网页头部

网页头部主要是企业 Logo 和一些快速链接，如关于我们、食品知识、网银在线支付等。除此之外还有导航菜单栏和搜索框等。

本实例中网页头部的效果如图 23-3 所示。



图 23-3　网页头部

实现网页头部的详细代码如下：

```
div id="header"> <a href="index.html"  class="logoMain"><img
src="img/woolworths-logo.png" width="230" height="57" /></a>
  <form class="hSearch" id="searchForm" method="post" >
    <fieldset>
    <label for="search">
    <input id="search" class="hSearchText" type="text" onfocus=
      "this.value='';" value="请输入" name="search query"/>
    <input class="hSearchGo" type="image" src="img/search-btn-go.gif"
      value="Go"/>
    </label>
    </fieldset>
  </form>
  <ul id="navSub">
    <li> <a href="#" >登录</a></li>
    <li><a href="#" >联系我们</a></li>
    <li><a href="#" target="new" >注册</a></li>
    <li class="end"> <a href="#" title="Large Font" onclick= "setActiveStyleSheet
      ('large'); return false;">放大</a> <a href="#" title="normal font"
      class="small" onclick="setActiveStyleSheet('default'); return false;">
      缩小</a> </li>
  </ul>
```

23.3.3　主体第一通栏

网页中间主体的第一通栏主要包括选购商品、在线支付、免费试吃、冷藏物流、速递直达、客户服务等，具体效果如图 23-4 所示。













图 23-4　主体第一通栏

实现主体第一通栏功能的具体代码如下：

```
<div class="container clearfix" id="wrapper">
  <div class="alternate" id="home">
    <div class="grid 1" id="sidebar">
      <ul>
        <li id="btn-whatsnew"><a href="#" ><span>选购商品</span></a></li>
        <li id="btn-specials"><a href="#" ><span>在线支付</span></a></li>
        <li id="btn-shop"><a href="#" ><span>免费试吃</span></a></li>
        <li id="btn-work"><a href="#" ><span>冷藏物流</span></a></li>
        <li id="btn-everyday"><a href="#" ><span>速递直达</span></a></li>
        <li id="btn-recipes"><a href="#" ><span>客户服务</span></a></li>
      </ul>
    </div>
```

23.3.4　主体第二通栏

网页中间主体的第二通栏主要是热销商品的展示，具体效果如图 23-5 所示。



图 23-5　主体第二通栏

实现主体第二通栏功能的具体代码如下：

```
      <div>
        <table width="930" height="310" border="0" align="center"
          cellpadding="0" cellspacing="0">
          <tr>
            <td width="930" height="310" align="center"><div class=pic show
              style="width:930px;">
```

```
            <div id="imgADPlayer"></div>
            <script type="text/jscript" language="javascript">
           PImgPlayer.addItem( "", "", "img/01.jpg");
           PImgPlayer.addItem( "", "", "img/02.jpg");
           PImgPlayer.addItem( "", "", "img/03.jpg");
           PImgPlayer.addItem( "", "", "img/04.jpg");
           PImgPlayer.addItem( "", "", "img/05.jpg");
           PImgPlayer.init( "imgADPlayer", 930, 310 );
             </script>
          </div></td>
      </tr>
    </table>
  </div>
```

23.3.5 主体第三通栏

网页主体的第三通栏主要是商品分类模块，具体效果如图 23-6 所示。



图 23-6　主体第三通栏

实现主体第三通栏功能的具体代码如下：

```
<div class="promotop grid" style="padding-top:10px;" >
    <h3><a href="#" > 梦幻棉花糖</a></h3>
    <hr/>
    <a href="#" > <img src="img/promo-comm-grants[1].jpeg~MOD=AJPERES&amp;
      CACHEID=24fd40004118e387a1d6e9f9a5cf1c57.jpg" border="0"  width="145"
      height="100"  /> </a>
    <p>棉花糖蓬松柔软，入口即溶，口味甘甜，深受很多年轻人的青睐</p>
    <p><a href="#" class="arrow">详细内容</a></p>
  </div>
  <div class="promotop grid" style="padding-top:10px;" >
    <h3><a href="#" >进口食品 尝鲜正当时</a></h3>
    <hr/>
    <a href="#" > <img src="img/FFM Annette 145x100.jpg~MOD=AJPERES&amp;
      CACHEID=e95e780041737cdab7a4bf5af93b836b.jpg" border="0"  width="145"
      height="100"  /> </a>
    <p>基于绝大多数进口食品的价格都高于市面上同类国产食品</p>
    <p><a href="#" target="_self" class="arrow">详细内容</a></p>
  </div>
  <div class="promotop grid" style="padding-top:10px;" >
    <h3><a href="#" > 美味体验：美国青豆买十送一</a></h3>
    <hr/>
```

```
    <a href="#" > <img src="img/145x100 Agricultural.jpg~MOD=AJPERES&amp;
      CACHEID=a92f600041b6e21db6f8f7e779ac7bf4.jpg" border="0" width="145"
      height="100"  /> </a>
    <p>本活动精选八款商品，分别是：美国青豆芥末味(小包装)、美国青豆芥末味(大包装)</p>
    <p><a href="#" class="arrow">详细内容</a></p>
  </div>
  <div class="promotop grid" style="padding-top:10px;" >
    <h3><a href="#" > 松脆好口感 方形威化饼</a></h3>
    <hr/>
    <a href="#"> <img src="img/freshMarketUpdatePromoTile.jpg~MOD= AJPERES&amp;
      CACHEID=a0e690804118e365a0a3e8f9a5cf1c57.jpg" border="0" width= "145"
      height="100" /> </a>
    <p>威化饼采用新鲜、纯正、支链淀粉多、黏性大的糯米为主料；先将糯米洗净、浸泡、晾干、舂
      粉</p>
    <p><a href="#" class="arrow">详细内容</a></p>
  </div>
  <div class="promotop grid" style="padding-top:10px;" >
    <h3><a href="#" > 泰国干果 营养健康新选择</a></h3>
    <hr/>
    <a href="#" > <img src="img/145x100 SWS.jpg~MOD=AJPERES&amp; CACHEID=
      df8e8900420d7c1d9a94fe2d0d22fd60.jpg" border="0" width="145"
      height="100"  /> </a>
    <p>花生滋养补益，有助于延年益寿，所以民间又称之为“长生果”。</p>
    <p><a href="#" target=" self" class="arrow">详细内容</a></p>
  </div>
  <div class="promotop grid" style="padding-top:10px;" >
    <h3><a href="#" > 开怀尝鲜“洋零食”</a></h3>
    <hr/>
    <a href="#"> <img src="img/145x100 question1.jpg~MOD=AJPERES&amp; CACHEID=
      30d0cd80422335298526ff2d0d22fd60.jpg" border="0" width="145"
      height="100"  /> </a>
    <p>只要你稍微留心一下，便会发现身边的进口食品专营店从稀少到常见，越来越多。</p>
    <p><a href="#" class="arrow">详细内容</a></p>
  </div>
</div>
<br />
<br />
<div>
```

23.3.6　网页底部

网页底部主要包括友情链接模块、快速访问模块等内容。相对比较简单，具体效果如图 23-7 所示。

图 23-7　网页底部

实现网页底部功能的具体代码如下：

```
<div id="quickLinks" class="container">
    <h3>快速导航</h3>
    <div class="grid 1">
      <h4><a href="#" >博客园</a></h4>
      <ul>
        <li><a href='#'> 查看最新</a></li>
        <li ><a href='#'> 写博客</a></li>
        <li ><a href='#'> 进入博客园</a></li>
      </ul>
      <h4><a href="#" >VIP 专区</a></h4>
      <ul>
        <li><a href="#" >VIP 会员登录</a></li>
        <li class=""><a href="#" >申请 VIP 会员</a></li>
        <li class=""><a href='#'> 订阅免费期刊</a></li>
        <li class=""><a href='#'> VIP 会员的优惠</a></li>
        <li class=""><a href='#'> VIP 会员帮助</a></li>
      </ul>
    </div>
    <div class="grid 1">
      <h4><a href="#" >儿童食品选购</a></h4>
      <ul>
        <li class=""><a href='#'> 婴幼儿食品</a></li>
        <li class=""><a href='#'> 1~3 岁儿童食品</a></li>
        <li class=""><a href='#'> 婴幼儿乳制品</a></li>
        <li class=""><a href='#'> 儿童乳制品</a></li>
        <li class=""><a href='#'> 儿童零食</a></li>
        <li class=""><a href='#'> 儿童饮料</a></li>
        <li class=""><a href='#'> 专家咨询</a></li>
      </ul>
    </div>
    <div class="grid_1">
      <h4><a href="#" >美食社区</a></h4>
      <ul>
        <li class=""><a href='#'> 进入社区</a></li>
        <li class=""><a href='#'> 最新动态</a></li>
        <li class=""><a href='#'> 专题报道</a></li>
        <li class=""><a href='#'> 讨论专区</a></li>
        <li class=""><a href='#'> 社区帮助</a></li>
      </ul>
      <h4><a href="#" >食品知识</a></h4>
      <ul>
       <li class=""><a href='#'> 食物的搭配</a></li>
        <li class=""><a href='#'> 美食营养学</a></li>
        <li class=""><a href='#'> 注意要点</a></li>
        <li class=""><a href='#'> 在线咨询</a></li>
      </ul>
    </div>
    <div class="grid 1">
      <h4><a href="#" target=" blank" >网站帮助</a></h4>
      <ul>
        <li><a href="#" target=" blank" >在线提问</a></li>
        <li><a href="#" target="_blank" >意见建议</a></li>
```

```
    </ul>
    <h4><a href="#" >加入我们</a></h4>
    <ul>
      <li><a href="#" target="_blank" >事业特色</a></li>
      <li><a href="#" target=" blank" >建店支持</a></li>
      <li><a href="#" target="_blank" >经营管理</a></li>
      <li><a href="#" target=" blank" >在线申请</a></li>
    </ul>
    <h4><a href="#" target=" blank" >网银在线支付</a></h4>
    <ul>
       <li><a href="#" target=" blank" >支付平台</a></li>
      <li><a href="#" target=" blank" >支付流程</a></li>
      <li><a href="#" target=" blank" >支付帮助</a></li>
    </ul>
  </div>
  <div class="grid 1">
    <h4><a href="#" >关于我们</a></h4>
    <ul>
      <li class=""><a href='#'> 关于公司</a></li>
      <li class=""><a href='#'> 关于团队</a></li>
      <li class=""><a href='#'> 联系我们</a></li>
      <li class=""><a href='#'> 社会责任</a></li>
      <li class=""><a href='#'> 展望未来</a></li>
      <li class=""><a href='#'> 公司新闻</a></li>
    </ul>
  </div>
  <div class="grid 1">
    <ul >
      <li><a href="#" target="_blank"  class="bold">意见建议</a></li>
      <li><a href="#" target=" blank"  class="bold">问题投诉</a></li>
      <li><a href="#"  class="bold">加盟通道</a></li>
      <li><a href="#"  class="bold">联系我们</a></li>
      <li><a href="#"  class="bold">人才招聘</a></li>
    </ul>
  </div>
  <div class="clear"></div>
 </div>
 <div id="footer">
  <p class="small">儿童食品网. 保留一切权利.</p>
 </div>
</div>
```



第 24 章 开发连锁酒店订购系统

本章介绍一个酒店订购系统的开发。这里将使用前面学习的 local Storage 来处理订单的存储和查询。该系统主要功能为订购房间、查询连锁分店、查询订单、查看酒店介绍等功能。通过本章的学习，用户可以了解在线订购系统的制作方法、使用 localStorage 模拟在线订购和查询订单的方法和技巧。

重点案例效果

千谷连锁酒店系统







立即预订 连锁分店 我的订单 关于千谷

订购专线：12345678

回首页 订单列表

删除订单 以下为您的订购列表：

房间楼层	是否带窗户	是否需接送
二楼	有窗口	需要接送
订购数量	客户联系方式	
26	123456789	

订购专线：12345678

24.1 连锁酒店订购的需求分析

需求分析是连锁酒店订购系统开发的必要环节。该系统的需求如下。

(1) 用户可以预订不同的房间级别，定制个性化的房间，而且还可以快速搜索自己需要的房间类型。

(2) 用户可以查看全国连锁酒店的分店情况，并且可以自主联系酒店的分店。

(3) 用户可以查看预订过的订单详情，还可以删除不需要的订单。

(4) 用户可以查看连锁酒店的介绍。

制作完成后的首页效果如图 24-1 所示。



图 24-1 连锁酒店订购系统的首页效果

24.2 网站的结构

分析完网站的功能后，开始分析整个网站的结构，主要分为以下 5 个页面，如图 24-2 所示。

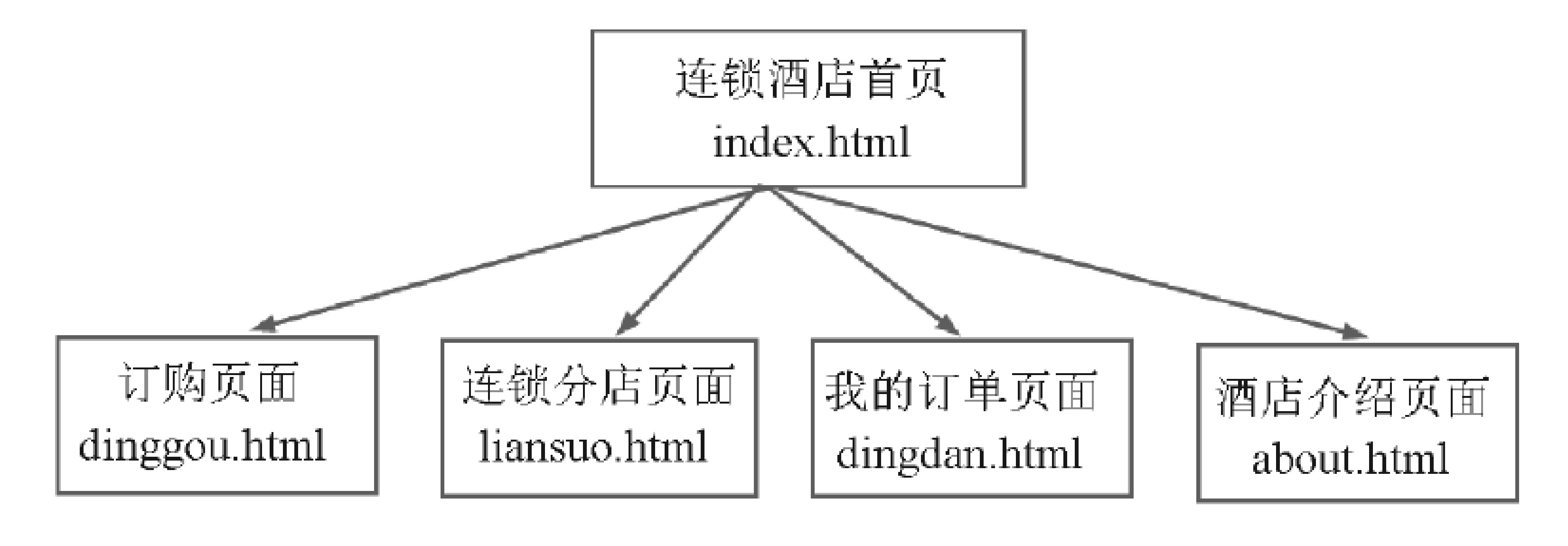


图 24-2 网站的结构

各个页面的主要功能如下。

(1) index.html。该页面的系统的主页面，主要是网站的入口，通过主页可以链接到订购页面、连锁分店页面、我的订单页面和酒店介绍页面。



(2) dinggou.html。该页面是酒店订购页面，主要包括 3 个 page，第一个 page 是选择房间类型；第二个 page 主要功能是选择房间的具体参数；第三个 page 是显示订单完成信息。

(3) liansuo.html。该页面主要显示连锁分店的具体信息。

(4) dingdan.html。该页面主要显示用户已经订购的订单信息。

(5) about.html。该页面主要显示关于连锁酒店的简单介绍。

24.3　连锁酒店订购系统的代码实现

下面来分析连锁酒店订购系统的代码是如何实现的。

24.3.1　设计首页

首页中主要包括 1 个图片和 4 个按钮，分别连接到订购页面、连锁分店页面、我的订单页面和酒店介绍页面。主要代码如下：

```
<div data-role="page" data-title="Happy" id="first" data-theme="a">
<div data-role="header">
<h1>千谷连锁酒店系统</h1>
</div>
<div data-role="content" id="content" class="firstcontent">
    <img src="images/zhu.png" id="logo"><br/>
    <a href="caigou.html" data-ajax="false" data-role="button" data-
      icon="home" data-iconpos="top" data-mini="true" data-
      inline="true"><img src="images/cai.png"><br>立即预订</a>
    <a href="liansuo.html" data-ajax="false" data-role="button" data-
      icon="search" data-iconpos="top" data-mini="true" data-
      inline="true"><img src="images/lian.png"><br>连锁分店</a>
    <a href="dingdan.html" data-ajax="false" data-role="button" data-
      icon="gear" data-iconpos="top" data-mini="true" data-
      inline="true"><img src="images/ding.png"><br>我的订单</a>
      <a href="about.html" data-ajax="false" data-role="button" data-
        icon="gear" data-iconpos="top" data-mini="true" data-
        inline="true"><img src="images/ding.png"><br>关于千谷</a>
</div>
<div data-role="footer" data-position="fixed" style="text-align:center">
  订购专线：12345678
</div>
</div>
```

其中 data-ajax="false"表示停用 Ajax 加载网页；data-role="button"表示该链接的外观以按钮的形式显示；data-icon="home"表示按钮的图标效果；data-iconpos="top"表示小图标在按钮上方显示；data-inline="true"表示以最小宽度显示。效果如图 24-3 所示。



图 24-3　链接的样式效果

其中页脚部分通过设置属性 data-position="fixed"，可以让页脚内容一直显示在页面的最下方。通过设置 style="text-align:center"，可以让页脚内容居中显示，如图 24-4 所示。



图 24-4　页脚的样式效果

24.3.2　订购页面

订购页面主要包含 3 个 page，主要包括选择房间类型 page(id=first)、选择房间的具体参数 page(id=second)和显示订单完成信息 page(id=third)。

1. 选择房间类型 page

其中选择房间类型 page 中包括房间列表、返回到上一页、快速搜索房间等功能。代码如下：

```
<div data-role="page" data-title="房间列表" id="first" data-theme="a">
<div data-role="header">
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">Back</a> <h1>房间列表</h1>
</div>
<div data-role="content" id="content">
    <ul data-role="listview" data-inset="true" data-filter="true" data-
filter-placeholder="快速搜索房间">
        <li>
             <a href="#second">
             <img src="images/putong.png" />
             <h3>普通间</h3>
             <p>24 小时有热水</p>
             </a>
             <a href="#second" data-icon="plus"></a>
        </li>
        <li>
            <a href="#second">
             <img src="images/wangluo.png" />
             <h3>网络间</h3>
             <p>有网络和电脑、24 小时热水</p>
            </a>
            <a href="#second" data-icon="plus"></a>
        </li>
        <li>
            <a href="#second">
             <img src="images/haohua.png" />
             <h3>豪华间</h3>
             <p>免费提供三餐、有网络和电脑、24 小时热水</p>
            </a>
```

```
            <a href="#second" data-icon="plus"></a>
        </li>
        <li>
            <a href="#second">
              <img src="images/zongtong.png" />
              <h3>总统间</h3>
              <p>24 小时客服、有网络和电脑、24 小时热水、免费提供三餐</p>
            </a>
            <a href="#second" data-icon="plus"></a>
        </li>
    </ul>
        </div>
<div data-role="footer" data-position="fixed" style="text-align:center">
  订购专线：12345678
</div>
</div>
```

房间列表页面的效果如图 24-5 所示。

页面中有一个 Back 按钮，主要作用是返回到主页，通过以下代码来控制：

```
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">Back</a>
```

房间列表使用 listview 组件，通过设置 data-filter="true"，就会在列表上方显示搜索框；通过设置 data-inset="true"，可以让 listview 组件添加圆角效果，而且不与屏幕同宽；其中 data-filter-placeholder 属性用于设置搜索框内显示的内容，当输入搜索内容，将查询出相关的记账信息，如图 24-6 所示。



图 24-5　房间列表



图 24-6　快速搜索房间

2. 选择房间的具体参数 page

选择房间的具体参数 page 的 id 为 second，主要让用户选择楼层、是否带窗口、是否需要接送、订购数量和用户联系方式，如图 24-7 所示。

这个页面的 Back 按钮的设置方法和上一个 page 不同，通过设置属性 data-add-back-

btn="true"实现返回上一页的功能，代码如下：

```
<div data-role="page" data-title="选择房间" id="second" data-theme="a" data-add-back-btn="true">
```

该页面包含选择菜单(Select menu)、2 个单选按钮组件(Radio button)、范围滑块(Slider)、文本框(text)和按钮组件(button)。

其中添加选择菜单(Select menu)的代码如下：

```
<div data-role="content" id="content">
    选择楼层:
    <select name="selectitem" id="selectitem">
      <option value="一楼">一楼</option>
      <option value="二楼">二楼</option>
      <option value="三楼楼">三楼</option>
    </select>
```

选择菜单的预览效果如图 24-8 所示。



图 24-7　选择房间　　　图 24-8　选择菜单

2 个单选按钮组的代码如下：

```
<fieldset data-role="controlgroup">
        <legend>选择是否带窗口：</legend>
            <input type="radio" name="flavoritem" id="radio-choice-1"
               value="有窗口" checked />
            <label for="radio-choice-1">有窗户</label>
            <input type="radio" name="flavoritem" id="radio-choice-2"
               value="无窗户"  />
            <label for="radio-choice-2">无窗户</label>
               <fieldset data-role="controlgroup1">
        <legend>选择是否接送：</legend>
            <input type="radio" name="flavoritem1" id="radio-choice-3"
```

```
                value="需要接送" checked />
            <label for="radio-choice-3">需要接送</label>
            <input type="radio" name="flavoritem1" id="radio-choice-4"
                value="无需接送" />
            <label for="radio-choice-4">无需接送</label>
```

单选按钮组的预览效果如图 24-9 所示。

使用<fieldset>标记创建单选按钮组，通过设置属性 data-role="controlgroup"，可以让各个单选按钮外观像一个组合，整体效果比较好。

范围滑杆的代码如下：

```
<input type="range" name="num" id="num" value="1" min="0" max="100" data-
highlight="true" />
```

范围滑杆的预览效果如图 24-10 所示。



图 24-9　单选按钮组

订购数量：
26

图 24-10　范围滑杆

文本框的代码如下：

```
<input type="text" name="text1" id="text1" size="10" maxlength="10" />
```

其中 size 属性用于设置文本框的长度；maxlength 属性设置输入的最大值。

文本框的预览效果如图 24-11 所示。

确认按钮的代码如下：

```
<input type="button" id="addToStorage" value="确认订单" />
```

确认按钮的预览效果如图 24-12 所示。



图 24-11　文本框

确认订单

图 24-12　确认按钮

3. 显示订单完成信息 page

显示订单完成信息 page 的代码如下：

```
<div data-role="page" id="third">
<div data-role="header">
```

```
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">回首页</a> <h1>订购完成</h1>
</div>
<div data-role="content" id="content">
<img src="images/ding.png" /><br>
<font style="font-size:20px;">感谢您选择我们酒店<br>
以下为您的订购房间信息：</font>
<p><div id="message" style="font-size:25px;color:#ff0000"></div>
</div>
<div data-role="footer" data-position="fixed" style="text-align:center">
  订购专线：12345678
</div>
</div>
```

订单完成信息页面的预览效果如图 24-13 所示。



图 24-13　订单完成信息

接收订单的功能是通过 JavaScript 来完成的，代码如下：

```
<script type="text/javascript">
 var orderitem = "orderitem";
 var flavor = "itemflavor";
var flavor1 = "itemflavor1";
 var num = "num";
 var text1 = "text1";
       $("#second").live('pagecreate', function() {
          $('#addToStorage').click(function() {
             localStorage.orderitem=$("select#selectitem").val();
               localStorage.flavor=$('input[name="flavoritem"]:checked').val();
                         localStorage.flavor1=$('input[name=
                            "flavoritem1"]:checked').val();
               localStorage.num=$('#num').val();
               localStorage.text1=$('#text1').val();
                  $.mobile.changePage($('#third'),{transition: 'slide'});
          });
       });
        $('#third').live('pageinit', function() {
               var itemflavor = "房间楼层："+ localStorage.orderitem+"<br>
```

```
是否带窗户："+localStorage.flavor+"<br>是否需接送："+localStorage.flavor1+"<br>
房间数量："+localStorage.num+"<br>客户联系方式：
"+localStorage.text1;
                    $('#message').html(itemflavor);
                   //document.getElementById("message").innerHTML= itemflavor
            });
</script>
```

其中$符号代表组件，例如$("#second")表示 id 为 second 的组件。live()函数为文件页面附加事件处理程序，并规定事件发生时执行的函数。例如，下面的代码表示当 id 为 second 的页面发生 pagecreate 事件时，就执行相应的函数：

```
$("#second").live('pagecreate', function() {…});
```

当 id 为 second 的页面确认订单时，将会把订单的信息保存到 localStorage。当加载到 id 为 third 的页面加载时，将 localStorage 存放的内容取出来并显示在 id 为 message 的<div>组件中。代码如下：

```
   $('#third').live('pageinit', function() {
                    var itemflavor = "房间楼层："+ localStorage.orderitem+"<br>
是否带窗户："+localStorage.flavor+"<br>是否需接送："+localStorage.flavor1+"<br>
房间数量："+localStorage.num+"<br>客户联系方式：
"+localStorage.text1;
                    $('#message').html(itemflavor);
            });
```

其中$('#message').html(itemflavor)的语法作用和下面的代码一样，都是用 itemflavor 字符串替代<div>组件中的内容：

```
document.getElementById("message").innerHTML= itemflavor;
```

24.3.3　连锁分店页面

连锁分店页面为 liansuo.html，主要代码如下：

```
<div data-role="page" data-title="全国连锁酒店" id="first" data-theme="a">
<div data-role="header">
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">回首页</a>
<h1>全国连锁酒店</h1>
</div>
<div data-role="content" id="content">
    <ul data-role="listview" data-inset="true">
                <li>
                    <a href="#" onclick="getmap('上海连锁酒店')" id=btn>
                      <img src="images/shanghai.png" />
                      <h3>上海连锁酒店</h3>
                      <p>咨询热线：19912345678</p>
                    </a>

                </li>
                <li>
                    <a href="#" onclick="getmap('北京连锁酒店')" id=btn>
```

```
                    <img src="images/beijing.png" />
                    <h3>北京连锁酒店</h3>
                    <p>咨询热线：18812345678</p>
                  </a>

              </li>
              <li>
                  <a href="#" onclick="getmap('厦门连锁酒店')" id=btn>
                    <img src="images/xiamen.png" />
                    <h3>厦门连锁酒店</h3>
                    <p>咨询热线：16612345678</p>
                  </a>

              </li>
          </ul>

</div>
<div data-role="footer" data-position="fixed" style="text-align:center">
  连锁酒店总部热线：12345678
</div>
</div>
```

连锁分店页面的预览效果如图 24-14 所示。



图 24-14　连锁分店页面

其中使用 listview 组件来完成列表的功能。通过链接的方式返回到首页，代码如下：

```
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">回首页</a>
```

24.3.4　查看订单页面

查看订单页面为 dingdan.html，显示内容的代码如下：

```
<div data-role="page" data-title="订单列表" id="first" data-theme="a">
<div data-role="header">
```



```
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">回首页</a><h1>订单列表</h1>
</div>
<div data-role="content" id="content">
<a href="#" data-role="button" data-inline="true" onclick="deleteOrder();">
删除订单</a>
以下为您的订购列表：
<div class="ui-grid-b">
  <div class="ui-block-a ui-bar-a">房间楼层</div>
  <div class="ui-block-b ui-bar-a">是否带窗户</div>
 <div class="ui-block-b ui-bar-a">是否需接送</div>

  <div class="ui-block-a ui-bar-b" id="orderitem"></div>
  <div class="ui-block-b ui-bar-b" id="flavor"></div>
 <div class="ui-block-b ui-bar-b" id="flavor1"></div>
 <div class="ui-block-c ui-bar-a">订购数量</div>
  <div class="ui-block-c ui-bar-a">客户联系方式</div>
 <div class="ui-block-c ui-bar-a"></div>
  <div class="ui-block-c ui-bar-b" id="num"></div>
  <div class="ui-block-c ui-bar-b" id="text1"></div>
</div>
</div>
<div data-role="footer" data-position="fixed" style="text-align:center">
  订购专线：12345678
</div>
```

查看订单页面的预览效果如图 24-15 所示。



图 24-15　查看订单页面

该页面的主要功能是将 localStorage 的数据取出并显示在页面上，主要由以下代码实现：

```
<script type="text/javascript">
$('#first').live('pageinit', function() {
    $('#orderitem').html(localStorage.orderitem);
    $('#flavor').html(localStorage.flavor);
       $('#flavor1').html(localStorage.flavor1);
    $('#num').html(localStorage.num);
       $('#text1').html(localStorage.text1);
});
</script>
```

通过单击页面中的“删除订单”按钮，可以删除订单，通过以下函数实现删除功能：

```
function deleteOrder(){
    localStorage.clear();
    $(".ui-grid-b").html("已取消订单!");
}
```

24.3.5 酒店介绍页面

酒店介绍页面为 about.html，该页面的主要代码如下：

```
<div data-role="page" data-title="全国连锁酒店" id="first" data-theme="a">
<div data-role="header">
<a href="index.html" data-icon="arrow-l" data-iconpos="left" data-
ajax="false">回首页</a><h1>千谷连锁酒店</h1>
</div>
<div data-role="content" id="content">

<img src="images/about.png" /><br>
<font style="font-size:20px;">千谷连锁酒店集团定位于全国连锁高级酒店的发展,完善的酒
店预订系统,让您预订酒店客房更加轻松快捷,是您出差、旅游的好选择。</font>

</div>
<div data-role="footer" data-position="fixed" style="text-align:center">
  连锁酒店总部热线：12345678
</div>
</div>
```

酒店介绍页面的预览效果如图 24-16 所示。



图 24-16 酒店介绍页面